Robust and Low-Cost Active Sensors by means of Signal Processing Algorithms

la Cour-Harbo, Anders

Publication date:
2002

Document Version
Publisher's PDF, also known as Version of record

Robust and Low-Cost Active Sensors by means of Signal Processing Algorithms

Ph.D. thesis

Anders la Cour-Harbo

Department of Control Engineering
Aalborg University
Fredrik Bajers Vej 7, 9220 Aalborg East, Denmark

8th August 2002

ISBN 87-90664-13-2
Doc. no. D-4562
August 2002

This thesis was typeset using LaTeX2$_\varepsilon$ in `report` document class.

Preface

This thesis is submitted as partly fulfillment of the requirements for the Doctor of Philosophy at the Department of Control Engineering at the Institute of Electronic Systems, Aalborg University, Denmark. The work has been carried out in the period August 1998 to August 2002 under the supervision of Professor Jakob Stoustrup and Associate Professor Tom S. Pedersen.

The subject of the thesis is identification and development of algorithmic methods for improving the performance and functionality of low-cost active sensors. The thesis is mainly a theoretical approach to this challenge as the aim has been to obtain generic results rather than application specific results. However, some effort has been invested in demonstrating that the results can indeed be applied to real world sensors.

The Ph.D. study is a part of the STVF-financed project OPTOCTRL at Department of Control Engineering. This project is about nonlinear and robust control of electro-mechanical systems with optical sensors, and was initiated in collaboration with Bang & Olufsen, Denmark. The purpose was to identify control and signal processing algorithms for increasing robustness in system with optical sensors.

This work is thus supported by the Danish Technical Science Foundation (STVF) Grant no. 9701481.

Aalborg University, August 2002
Anders la Cour-Harbo

Abstract

The primary purpose of this thesis is to identify methods for improving the performance and providing additional and new functionality in active sensors. The principal idea for achieving this is to introduce integrated circuits, in particular on-chip computers, as a standard component. This enables the use of signal processing algorithms for providing the desired performance and functionality.

Improved performance is at this point in time synonymous with increased robustness, small size, and low cost. A reduction in size is a natural consequence of using integrated circuits and in large quantities the cost is in general reasonable. Though it can be a significant technical challenge to reduce size and cost of the hardware, the main concern in this thesis is robustness and functionality by means of signal processing algorithms, and the presence of a on-chip computer is simply a prerequisite for the methods suggested.

Introducing new functionality in sensors means providing the sensor with the ability to respond to input in a previously unseen way. An example is an automatic door sensor which can detect whether people are walking through or by the door, and only open the door in the former case. As an example of an interesting and useful functionality this thesis presents the first steps towards a sensor capable of determining the position of an object in three dimensions.

The thesis is divided into three parts. The first part is dedicated to a presentation of an algorithm for increasing the performance of active sensors, in particular for increasing the robustness. The second part presents methods for introducing determination of spatial position as a new functionality in a sensor. The third part is a presentation of a series of mathematical subjects which are relevant for the methods discussed in the first part.

The aim of the thesis is to provide signal processing methods for real applications of active sensors. This is achieved by reporting on a number of results of mainly mathematical nature, and subsequently showing how to employ those methods in real applications. While the former obviously presupposes some knowledge of mathematics, the latter typically requires skills in electronic engineering.

Contents

1 Introduction to Thesis

1.1 Focus and Purpose

The primary purpose of the present thesis is to identify methods for improving the performance and providing additional and new functionality in active sensors. The principal idea for achieving this is to introduce integrated circuits, in particular on-chip computers, as a standard component. This step will greatly increase the potential of active sensors, both in terms of improved performance and increased functionality. The reason is that on-chip computers allow for complex processing and decision-making which is virtually impossible to achieve with traditional analog circuitry.

Improved performance is at this point in time synonymous with increased robustness, small size, and low cost (see the Sensor Foresight Report [73]). A reduction in size is a natural consequence of using integrated circuits and in large quantities the cost is in general reasonable. Thought it can be a significant technical challenge to reduce size and cost of the hardware, the main concern in this thesis is robustness, and the presence of a on-chip computer is simply a prerequisite for the methods suggested.

As the title of the thesis suggests the primary tool for increased robustness is signal processing algorithms. The background of the author is mathematics, in particular functional analysis and operator theory, and the suggested methods bear witness of this as the mathematical aspect is predominant throughout the thesis. The author believes that this approach to the challenge provides a more generic solution which applies to active sensors in general rather than just any specific type on which the methods happen to be tested or implemented.

While traditional methods such as modulation out of base band and resonance filters sometimes make a good addition to the suggested algorithm no attempt has been made to develop such methods further.

The first part of the thesis, which is on increasing the robustness, is focused on developing methods which can (eventually) be implemented in on-chip computers. Although the actual implementation is not discussed in the thesis it is recognized that any suggested method should be suitable for implementation in low-cost signal processing hardware. This means that the signal processing algorithms must obey constraints on computational load, programmable complexity, and numerical stability.

Introducing new functionality in sensors means providing the sensor with the ability

to respond to input in a previously unseen way. An example is an automatic door sensor which can detect whether people are walking through or by the door, and only open the door in the former case. In this thesis a sensor capable of determining the position of an object in three dimensions is presented. As mentioned above the focus is on the theoretical and mathematical aspects of this functionality.

1.1.1 Background

The Ph.D. study started in August 1998 as a response to an interest of Bang & Olufsen to investigate the potential of combining wavelets and digital signal processing in active sensors. B&O wanted a generic solution to the problem of detecting an object (a feature in some of their products), because it is a surprisingly difficult task to design a robust, versatile, and low-cost detection system. The original idea from B&O was to employ digital signal processing, in particular to use wavelets for 'doing the signal processing'. Since the BeoSound Ouverture (CD player, see Section 3.4) was in mind as test application a number of constraints existed from the very beginning. Especially, the response time, cost, and computational power was limited. From early on the focus was therefore on providing a generic and robust detection system which can easily be tailored to obey given performance requirements.

1.1.2 Prerequisites

To fully appreciate the entire thesis the reader should have knowledge in a number subjects within mathematics and electrical and electronic engineering. The thesis is based on mathematical reasoning throughout, and especially Part II and III require the reader to have a certain mathematical level. The engineering aspects are predominant in Part I.

It is useful to have knowledge in the following mathematical disciplines: Linear algebra, Euclidean geometry, Fourier analysis, functional analysis, basic operator theory, wavelet theory, and basic probability and statistics. The engineering skills that make the reading easier are: digital and analog filters, filter design, basic electric circuits, Fourier analysis (from an engineering point of view), and discrete-time signal processing in general.

1.2 Content of Thesis

The thesis is divided into three parts. The first part is dedicated to presenting an algorithm for increasing the performance of active sensors, in particular for increasing the robustness. The second part presents methods for introducing a new functionality in a sensor. This new functionality is determining spatial position of an object. The third part is a presentation of a series of mathematical subjects which are relevant for methods discussed in the first part. In this section each part and each chapter is briefly introduced.

Chapter 2: Towards Intelligent Sensors is a short presentation of the author's point of view on the general state of the sensor industry and market in relation to research and development of new and more intelligent sensor, in short the context in which the present thesis should be regarded. The chapter also includes a description of what the author believes to be contributions to the development of active sensors.

1.2.1 Overview of Part I: Channel Gain Measurement

Part I of this thesis is focused on improving the performance of active sensors rather than adding new functionality (this is the subject of Part II). In particular, Part I is focused on the ability of the sensor to function in many different environments. This is primarily a question of robustness. The core of Part I is the channel gain measurement (CGM) algorithm which is a suggestion for a generic method for obtaining the gain in an unknown channel. A channel can be anything from electric wires to a path through water or air. Part I is divided into four chapters; an introduction to active sensor technology, a thorough presentation of the suggested algorithm, a series of results when applied to real applications, and a discussion of the current and future work.

Chapter 3: Introduction provides the background for the suggested CGM algorithm. This is primarily a presentation of the the concept of active sensors which includes the active sensor technology, applications areas, and sensor performance parameters.

Chapter 4: Methods for Measurement of Channel Gain is a quite elaborate description of the suggested CGM algorithm. First, two particular embodiments of the algorithm is presented. Then the algorithm in its entirety is presented. It consists of a number of steps which are also discussed individually. The main concern throughout the algorithm is robustness, and handling of noise thus becomes an important issue. A significant part of Chapter 4 is about recognizing, estimating, and removing noise. Another important issue is algorithmic and computational complexity, and emphasis is put on keeping the complexity at a level suitable for low-cost signal processing hardware.

Chapter 5: Results is mainly a series of applications of the methods presented in Chapter 4. The various steps in the CGM algorithm is applied to real world signals, and the results are evaluated. Chapter 5 is focused on the ability of the algorithm to recognize, estimate, and remove noise, i.e. to what extent robustness can actually be achieved.

1.2.2 Overview of Part II: Spatial Position

The second part of the thesis introduces a new functionality in low-cost sensors. The idea is a sensor capable of determining position of object in three dimension using only channel gain measurements. The methods presented in this part are dedicated to the problem of mapping a series of CGMs made on a reflecting object into a three dimensional position of the object. An algorithm for implementing such a mapping is mathematically far more

complicated than the algorithm presented in the first part, and thus the focus in this part is on functionality, and not on robustness, response time, or the cost.

The first chapter discusses various methods for implementing such a mapping. One of these methods is investigated in more detail in the following chapter. Independently of the mapping method it is necessary to have a qualitative description of how the channel gain varies as a function of the position of the object. This is investigated in the third chapter of this part. Finally, the results chapter presents what has been achieved so far.

Chapter 6: Methods for Determining Spatial Location introduces the concept of spatial position in the context of low-cost sensors, and present the result of applying a neural network to the problem of mapping CGM information into position information.

Chapter 7: Geometric Solution based on Intersections of Spheroids is one of the mathematically appealing methods for mapping CGMs to spatial position is a geometrical modeling of the 'emitter, receiver, reflecting object' setup. With this approach a completely analytical mapping can be constructed. The downside is that the complexity of the model is surprisingly high. A number of assumption has been imposed to reduce the complexity, but the price paid is reduced accuracy. However, some important results have been obtained, nonetheless.

Chapter 8: Modeling Reflection Maps introduced a model to describe how the object reflects the signal. The modeling of the setup in the previous chapter was based on a simple reflection model. In this chapter the reflection map for a infrared emitter/receiver pair is modeled as well as measured. This reveals some surprising effects which the simplified model in Chapter 7 does not account for.

1.2.3 Overview of Part III: Wavelet and Rudin-Shapiro Transforms

The third part of the thesis is focused on purely mathematical subjects related to the wavelet and Rudin-Shapiro transform. The subjects discussed here all originate in problems that have arisen in the work with active sensors.

First, the problem of a proper handling of the edges when wavelet transforming finite signals is treated in detail in the first two chapters. In active sensors the signals are often quite short and thus the edge effect becomes an important factor. Note that the wavelet transform itself is not presented or discussed as the reader is expected to have some knowledge of the transform. Readers interested in learning about the transform are referred to the vast amount of literature on the subject.

Second, the Rudin-Shapiro transform is presented. Since this transform is virtually unknown compared to the wavelet transform it is introduced in some detail. This transform has a series of properties which makes it very useful for providing robustness in low-cost active sensors.

Chapter 9: The Problem of Finite Signals presents a number of standard solutions to the problem of wavelet transforming finite signals. This subject is important in the context

of this thesis as the wavelet packet transform is suggested as a possible component of the CGM algorithm. The constraints on the algorithm means that it is often necessary to transform quite short signal, which in turn means that it is important to have a proper handling of the edges.

Chapter 10: Moment Preserving Edge Filter introduces a more complicated solution to the problem of finite signals. The solution focuses on preserving a particular polynomial-related property that wavelets has on the real line, when the transform is restricted to an interval. The moment preserving edge filters are well suited for some of the typical noise that occurs in active sensors. A previously unreported, and yet significant, instability issue of this construction is also discussed.

Chapter 11: The Rudin-Shapiro Transform is an introduction to the Rudin-Shapiro polynomials, sequences, and transform, and to the concept of flat polynomials. A number of useful properties of the RST is derived, and a fast implementation of the transform is presented. This chapter includes a brief review of the history of flat polynomials.

Chapter 12: Linear Transform of the Rudin-Shapiro Matrix reports on a method for determining the impact of applying block diagonal linear transforms to RS sequences prior to RS transformation. This is relevant in cases where RS sequences are denoised after transmission in noise environments.

Chapter 13: Discussion and Future Work wraps up the thesis by concluding and discussion the presented algorithms, methods, applications and theory. A list of future work is also given. This chapter includes a discussion of the the necessity for signal processing algorithms in low-cost active sensors in comparison to traditional solutions.

1.2.4 Overview of Appendices

Appendix A: Basic Properties of the Wavelet Transform lists a series of well-known properties of the wavelet transform. This appendix is included solely to support the presentation of moment preserving filters in Chapter 10.

Appendix B: Extra Lemmas, Expressions, and Figures contains material which on the one hand fits poorly in to context, but on the other hand is useful nonetheless.

Appendix C: Moment Preserving Edge Filter in Matlab prints four functions which generate all the necessary components for performing filtering with moment preserving filters. The Matlab code reproduces the derivations and calculations presented in Chapter 10. These functions are printed in the thesis as the author has not been able to obtain any kind of computer implementations (or indeed any pre-calculated filter coefficients) and thus had to implement the edge filter construction from scratch.

1.3 Contributions

The following list briefly presents what the author claim is the contributions of this thesis.

Systematic method for increasing the robustness of low-cost active sensors
The main contribution of the thesis is the algorithm for measuring channel gain. The algorithm consists of a number of steps which are based on more or less well-known theory. Except for the fast implementation of the Rudin-Shapiro transform all the methods throughout the algorithm has been reported previously. However, the composition of these step into a single, flexible algorithm is believe to be new. That is, the concept systematizing the algorithm by using transforms and inverse transforms for signal modulation combined with tailor-made denoising is believe to be new in the context of low-cost active sensors. All of Part I is dedicated to this algorithm.

The concept of a 3D sensor based on channel gain measurements
Since the CGMs of Part I is in some sense relative measures of distance, it is natural to use them for determining the spatial position of an object. While a number of positioning systems exists, there are none, to the best of the author's knowledge, which are based solely on measurements by means of low-cost diodes of reflected intensities and with the sensors located in a two-dimensional plane. The thesis thus contributes to the list of sensor functionality with a physically simple and low-cost method for determining spatial position.

Geometric modeling of a simplified 3D sensor
One of the methods for mapping a set of CGMs to a spatial position is a model of the sensor setup. In the thesis a model based on geometric observations is presented. Although the assumptions are simplified a number of interesting results are reported. Among the question addressed is the expected complexity of a sufficiently accurate model, and the number and optimal position of sensors in the 2D plane.

Modeling of the reflection map for an 'emitter, receiver, reflection object'-setup
Any modeling of a sensor setup for determining spatial position must rely on a reflection map for the object. The thesis presents a measured reflection map for infrared diodes, and a model for predicting the reflection map. The measured reflection map exhibits an interesting and important non-symmetric characteristics which is replicated fairly well by the model. Together with the above two items this contributes to the understanding of how to construct a 3D sensor.

Implementation of moment preserving edge filters for the wavelet transform
A method for constructing and applying moment preserving edge filters was reported by Cohen et al. [22]. This thesis contributes by reporting on a numerical stability issue in the construction, and by providing MATLAB code for generating the filters for any given orthogonal wavelet filter.

A fast implementation of the symmetric Rudin-Shapiro transform
The symmetric Rudin-Shapiro transform has previously been reported by Byrnes [13], but no $N \log N$ implementation of the transform has been explicitly demonstrated before. A unified presentation of the most important properties of the symmetric transform is also given. The chapter on the RST also contains some new conjectures which has no particular bearing on the thesis; they just emerged when the author was investigating the RST.

Some Results on the Dyadic Structure of the Rudin-Shapiro Transform
The necessity to rectify the changes made to a RS sequence which has undergone polynomial denoising have led to some results on the dyadic structure of the RST, including a simple prediction of the impact of a block diagonal linear transform applied to a RS sequence prior to RS transformation. The entire Chapter 12 is a contribution to the theory of RS polynomials.

1.4 Acknowledgements

This thesis would not have been without the assistance of a series of very helpful people. I do appreciate the help and assistance that I have received, and I would like to acknowledge this by briefly mentioning how each person have assisted.

Palle Andersen and Tom S. Pedersen, associate professors at the local department, have assisted in designing the electrical circuits used in test setups described in Chapter 5. Pedersen has also acted as extra supervisor. Henrik Fløe Mikkelsen from Bang & Olufsen is the main contact to B&O. He came up with the initial problem and ideas for this Ph.D. study. Also, B&O has provided various equipment for testing purposes.

A part of a Ph.D. study is a longer stay at another institution. Lars F. Villemoes, previously associate professor at KTH, Stockholm, (he is now with Coding Technologies) invited me to a six month stay at Department of Mathematics, KTH. I am thankful to Villemoes for suggesting the use of Rudin-Shapiro polynomials, and for a series of discussions on various wavelet-related subjects. I would also like to thank Jan-Olov Strömberg, professor in computational harmonic analysis, KTH for a receiving me at KTH, and for inviting me to a quite interesting five day course by professor David Donoho. At KTH I was also fortunate to meet Harold Shapiro, professor emeritus, who incidentally had an office right across the hallway. Discussions with him helped me gain insight on the subject of flat polynomials, and to put Rudin-Shapiro polynomials in a historical context. He also gave me a copy of his 1951 master's thesis (see Section 11.2).

I would also like to acknowledge the influence of my former supervisor at the Department of Mathematics professor Arne Jensen. My interest in engineering science, particularly signal processing, is due to him. He suggested that I completed my master's degree at another engineering department rather than at the mathematical department (which incidentally let to a Ph.D. study at an engineering department), and he has on several occasions asked my to join him in lectures, and also in the writing of a book [45], all in the

spirit of bridging the classical gap between mathematicians and engineers. Thanks are also due to Lasse Borup, currently Ph.D. student at Department of Mathematics for being helpful in answering mathematical question whenever needed.

The inverse problem approach to estimation of model accuracy in Chapter 8 originates in a ph.d. course I attended in 1999. The lecturer was Per Christian Hansen, professor of scientific computing at Technical University of Denmark. He provided the software needed for computing the regularized solutions to the inverse problem, and he also gave a number of useful suggestions for the textual presentation.

During the Ph.D. study I was also fortunate to have several discussion with a number of people from Carlo Gavazzi Industry, in particular Claus Bo Jensen, former R&D director, and Per Thorsen from the R&D department. CGI is a manufacturer of many types of sensors, particular the types which are interesting in the context of this thesis. The relation to CGI have been very helpful in understanding the needs and current challenges faced in the sensor industry.

The prototype used in the second test setup in Chapter 5 was financed and built primarily by LEGO. The persons behind this initiative is Carl Erik Skjølstrup and Asbjørn Leth Vonsild from LEGO Engineering. This test setup has been helpful in recognizing some of the challenges faced when building 3D sensors and object recognition sensors.

Finally, I am very grateful to Jakob Stoustrup for taking great interest in my work and always being enthusiastic about my results, for respecting for my ideas and methods, and yet making sure they met a proper scientific standard, for his own sometimes ingenious, sometimes crazy ideas, and for support and commitment throughout my almost four years of Ph.D. study. I could not have asked for a better supervisor.

Towards Intelligent Sensors

2

Throughout the past 50 years of development in the computer industry the ratio between computational power and size has increased tremendously and continuously. During the past few years the cost and size of a reasonably powerful computer (1-20 MIPS) has reached a level which allows for implementation in sensor products that has a total production cost as low as 5 EUR. Computational power in that order of magnitude provides sufficient means for moderately complicated signal processing operations, and thus allows sensors to have functionality which it is almost impossible to provide with traditional analog sensor technology.

The sensor industry has far from exploited the full potential of this development. Although virtually all sensor markets are growing faster than most other markets, and are predicted to grow in the years to come, the sensor industry is rather conservative regarding new technology. While the advances in computers have found their way to turn key sensor solutions, where one of the competing factors is functionality, the individual (low-cost) sensor units are still rather basic throughout the sensor industry [73]. This is probably due to the fact that the main competing edge is cost and not functionality. Since the contribution margin of low-cost sensors are very small the radical change of design needed to introduce signal processing easily jeopardizes any advantage a sensor manufacturer might have.

2.1 The Next Generation

The sensor technology is constantly being developed, and the concept of a next generation sensor is in this respect somewhat fuzzy. Nevertheless, the author of this thesis believes that it makes perfect sense to discuss ‘the next generation’ of sensors. Especially as a step in the direction of more intelligent sensors. The introduction of on-chip computers is a major and inevitable step (as argued below) in the evolution of sensors, and this development will bring new functionality into existing sensors as well as it will bring completely new sensor types. A next generation sensor is equipped with significant computational power, but is not more expensive than the current generation. However, the next generation of sensors is not intelligent in the sense of rational decision making, but they employ state of the art signal processing algorithms to achieve a significant step in areas such as robustness and flexibility.

2.1.1 Why Intelligent Sensors?

So why is it interesting to research methods that provides new functionality, or just improve performance of existing functionality, in low-cost sensor units? Why would a sensor manufacturer be interested in the next generation if the market prefers traditional technology and functionality? Why take a significant technological risk if the competing edge is negligible? The answer to these question is threefold.

Firstly, the technological risk in intelligent sensors is almost entirely on the algorithms. This is because the hardware has received (and still are receiving) almost all the attention in terms of research and development. While the sensor market is conservative in respect to new functionality and fundamental new sensing technologies, there is constantly a demand for increased efficiency at lower costs. To meet these demands the sensor industry responds by a continuous development of the sensor hardware components. And the technological risk of employing digital solutions altogether is also quite small; the DSP technology is mature and well-established, and the potential of on-chip computers has been demonstrated in many other fields (like cellular phones).

Secondly, the sensor industry is indeed beginning to employ on-chip computers in low-cost sensors, primarily in the form of microprocessors. Although this type of on-chip computers usually offers a very limited computational power, and often just replaces a number of discrete electrical components rather than adding functionality or even improved performance, it is still a step in the direction of more intelligent sensors. In particular, it is introducing the concept of digital signal processing in sensors.

And thirdly, the demands of the sensor market for better and cheaper sensors are continuously increasing, and it becomes still more difficult to achieve a competing edge with traditional sensor technology.

While the competing edge of traditional sensors is cost, mainly, it is the author's belief that the next generation of sensors will compete on functionality. This is because it is not likely that the production cost of a sensor in general will be reduced (much) by employing digital technology. But this does not mean that the next generation of sensors will be pouring into the market. The process will be slow, and the traditional sensors will have a major part of the market for many years to come. However, this should not discourage one from doing research in the next generation, as the three arguments above demonstrate.

The need for intelligent sensors is also expressed in the Sensor Foresight Report from Sensor Technology Center, Denmark [73]. This international survey among 174 companies concludes:

> *"Some general technological key features have been identified: Low price, small size, robustness, dispensability, and the ability to self-calibrate. Future sensors are expected to be integrated systems with multiple applications."*

To fulfill this ambition it is necessary to introduce algorithm which will bring sensors much closer to being intelligent than they are today.

2.2 Contribution of the Thesis

There are many issues to be addressed when developing more intelligent sensors, and it is impossible to take the step from traditional sensors directly to highly intelligent sensors. The process will be long and slow, and consists of a great number of small steps and a fewer larger steps. Introducing digital technology is one of those steps.

The main purpose of the present thesis is to contribute to this process. Part I of the thesis presents a method for improving the robustness of the fundamental principle in active sensors. This is a contribution on a low level in the sensor architecture (see Section 3.1.2 on the sensor layer model). Part II presents the idea of a sensor which by very simple hardware is capable of determining the spatial position of a passive object. This is a contribution on a high level in the sensor architecture. Part III addresses some of the mathematical issues which is relevant in Part I. Although some of the theory presented in Part III is a contribution of purely mathematical nature, it still applies to relevant problems in the field of active sensors.

2.2.1 Robust Channel Gain Measurements (Part I)

This thesis contributes to the development of the signal and control layers in the sensor architecture by suggesting means for providing the next generation of sensors with sufficient intelligence to achieve a significant degree of robustness. This means making the sensor behave reasonably and reliably in many different situations. The lack of robustness is the Achilles' heel in many existing sensors (for instance an extreme disturbance will typically make a sensor behave unpredictably), and there is indeed a lot of room for improvement of the robustness. Two important aspects of robustness is advanced signal processing and decision making. The former is required to handle localized (in time, frequency, or some other domain) noise occurrences and to provide information for the latter, which is about making decisions based on the current state of the environment of the sensor. Both aspects are handled much more elegantly by on-chip computers than analog electronics, and DSP technology thus becomes an vital part of any method to increase the robustness.

But the signal processing hardware itself is obviously not enough; the intelligence is provided by software implemented algorithms. While the software is merely the means for a hardware appropriate description of the intelligence (the author do realize that an efficient software implementation can be a challenge in itself, but that is not within the scope of this thesis), the mathematics and the algorithms comprising the intelligence are the real challenges. Consequently, the contribution from this thesis is not software, but a description of how to design algorithms which implemented as software in a sensor will provide a certain degree of intelligence.

This is not to say that the suggested algorithm does not take into account any other properties than robustness. The importance of keeping the production costs low is acknowledged by the fact that the suggested algorithm providing increased robustness is easily implemented and behaves in a stable manner in low-cost signal processing hardware.

The algorithm has been tested in a number of prototype applications. A total of four different setups have been used, and each step in the algorithm has been applied in at least one of the setups. As a supplement to computer simulations the implementation of the algorithm in actual real-time applications have proven useful in developing some of the steps, and in realizing what measures are needed to ensure the robustness in real applications.

2.2.2 Spatial Position (Part II)

The method presented in Part I of the thesis forms the basis for the 3D sensor presented in Part II. This sensor is able to determine the position of an object in three dimensions by combining the information from several emitter/receiver pairs. While obtaining the information is relatively easy (once the method of Part I is available) it is rather difficult to map this information into a position in 3D. A number of options is discussed, and the geometrical approach is examined in detail. However, this examination is mainly of theoretical nature as the geometrical approach has not been tested in a real setup.

An alternative to the 3D sensor is an 'object sensor', which means converting the obtained channel gain information into object-type (with a priori known position) rather that object-position (with a priori known object). Doing both would of course be very interesting, but that is simply too ambitious at this point in time. In cooperation with LEGO and the WAVES project a prototype of an object recognition sensor has been constructed. This prototype demonstrates some of the challenges in combining several sensors and thereby creating new functionality. The object recognition functionality is not discussed in this thesis.

2.2.3 Mathematics for Signal Generation and Processing (Part III)

The two main issues of Part III is proper wavelet transformation of finite signals and the Rudin-Shapiro transform. The wavelet transform is a well-established mathematical tool for signal processing, and the focus is therefore on applying it to finite signals. This is particular interesting in the context of active sensors, where the real-time requirements often allows only relatively short signals to be obtained for transformation. There exists a series of methods for applying the WT to finite signals. One of the more interesting methods (in this context as well as many others) is discussed in detail, and a previously unreported numerical stability problem of the methods is discussed.

The second contribution of Part III is the Rudin-Shapiro transform. While the mathematical idea behind this transform dates back to the middle of the 20th century, and the transform itself is more than 10 years old, the efficient implementation of the transform is believed to be new. This is also true for some of the properties of the RST reported in Part III. The RST has turned out to be a very efficient tool for designing the signals to be used in the algorithm of Part I, and has consequently been implemented in all but one of the test setups.

Part I

Channel Gain Measurement

3 Introduction

This introduction presents a series of concepts, properties, and methods which are relevant in the field of active sensors and in the context of this thesis. Although this introduction is not a prerequisite for reading the following chapters (which to a large extent are of a mathematical nature), it is necessary for a deeper understanding of the reasons for some of the actions taken, choices made, and conclusions drawn in Chapter 4 and 5 on methods and results.

The first section deals with active sensor technology primarily from a conceptual point of view. The basics of an active sensor is presented, followed by the sensor level model. Section 3.1 provides an understanding of the context in which the present thesis is written.

The remaining sections 3.2 through 3.5 introduces various aspects of active sensors.

First a series of sensor applications are presented in Section 3.2 to give an impression of the usefulness and diversity of active sensors. Then in Section 3.3 the most common traditional methods for active sensing are presented. This is followed by presentation in Section 3.4 of the one application, the BeoSound Ouverture from Bang & Olufsen, which initiated the authors Ph.D. study and thus this thesis. Finally, Section 3.5 lists some important sensor performance parameters.

3.1 Active Sensor Technology

An active sensor is a sensor which performs sensing actively. It uses an emitter to influence the environment in order to cause a reaction which is measured by a receiver. In comparison, a passive sensor employs the receiver only.

This section introduces this concept. That includes a description of the physical construction, i.e. the most common and basic parts, in Section 3.1.1, and a conceptual description in Section 3.1.2 and 3.1.3. The conceptual description is advantageous in the context of this thesis where the main improvements are made by means of algorithms rather than hardware.

Finally, a very brief presentation of the most common sensing principles is given in Section 3.1.4.

3.1.1 Basic Parts in an Active Sensor

An active sensor consists basically of seven parts: The emitter with related electronics, the receiver with related electronics, the signal generator, the signal processor, and the operations part. Often two or more of these parts are integrated in the same electrical component, and often all the parts are housed in the same casing. The sensing process starts with the signal generator and ends with the operations part. In the some active sensors the loop is closed by a feedback from the operations part to the signal generator. The seven parts are illustrated in Fig. 3.1. The seven parts each has a specific purpose,

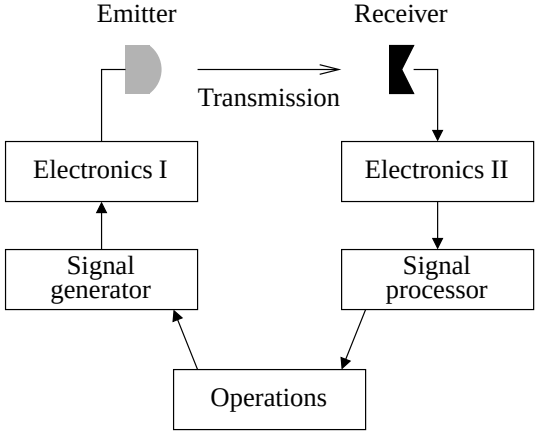


Figure 3.1: The basic components of an active sensor.

which are listed below.

Signal generator The process of generating a signal suitable for transmission consists of two steps; this and the following. The purpose of this step is to produce a signal with the desired mathematical properties such as localization in time and frequency. The signal generator can be anything from a constant to the result of a advanced algorithm in a DSP. Note that the outcome of this step is normally not very well suited to drive the emitter (the current and/or voltage level differs from the specifications for the diode).

Electronics I The purpose of the electronics between signal generator and emitter is to convert the signal into an analog signal suitable for the emitter. This includes D/A conversion, translation and scaling to fit the dynamic range of the emitter, and possibly modulation. This step is often implemented by discrete electrical component on a PCB. Sometimes the signal is binary (on/off) in which case the DAC can be omitted (which is often the reason for choosing a binary signal).

Emitter The emitting component obviously conforms with the choice of signal type (light, sound, etc.). The emitter might be capable of transmitting at many different

frequencies and amplitudes (like an ordinary loudspeaker) or at very specific frequencies and amplitudes (like a laser diode). The emitter is rarely integrated with the electronics, since the process of emitting a signal is often fairly straightforward, and not particular susceptible to disturbances such as cross talk.

Receiver The receiver must also conform with the type of transmitted signal, and must be able to receive any transmission from the emitter. Occasionally, the receiver is chosen to be sensitive only at a very limited range of frequencies of the carrier wave (like a radio is sensitive in a narrow (although adjustable) frequency band) to eliminate more noise. This is because the process of receiving a signal is often quite sensitive to external disturbances, including cross talk. This is in turn due to the output from the receiver components, which is often very weak and thus requires high amplification prior to A/D conversion.

Electronics II This amplification is handled by the electronics between receiver and signal processor. The high sensitivity means that this part of the sensor must be carefully designed. For that very reason, it is not uncommon to have receiver and amplifier integrated in one component. The A/D conversion is also considered a part of the electronics, but it is rarely integrated with the amplifier. In some cases the ADC is a separate component, in other cases it is an integrated part of the signal processor. In most sensors employing on-chip computers the ADC is a necessary component.

Signal processor The purpose of the signal processor is to determine the state of the received signal. In many cases this amounts to determining whether the emitted signal is present in the received signal or not. The signal processor part can be a simple threshold or comparator in analog electronics, and it can be a carefully devised and extensive examination of the signal in a powerful DSP. In any case the signal processor produces a result which is fed to the operations part.

Operations This final part has a number of functions for controlling the sensing process and providing the output from the sensor unit. One of the functions is to make the sensor respond to whatever happens to the signal during transmission. That means providing the output from the sensor unit to the mechanism or device which the sensor is connected to. Another function is to change the parameters of the various parts of the sensor based on the current state of the received signal. The operations part ranges from being not present at all to controlling every aspect of the sensor, including the electronics.

3.1.2 Sensor Level Model

In the introduction to the concept of intelligent sensors in Chapter 2 the contribution of the present thesis was said to be on two different levels. This was meant in a quite specific manner. An active sensor can be regarded as consisting of eight levels. Each level has a distinct task in the sensing process, and each level provides resources and information for the levels above it. These levels are shown in the sensor level model in Fig. 3.2. It is important to note that the first five levels in this model is closely related to (but not equal

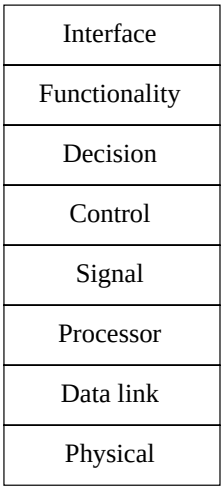


Figure 3.2: Sensor level model

to!) the basic parts of an active sensor described in the previous section, and 'orthogonal' to the sensor parameters described in Section 3.5. Note also that not all the levels in the model are present in all active sensors.

The sensor level model is an abstraction of the sensing process which makes it easier to understand the different types of contributions to the development of more advanced sensors. The following list describes each level and its relation to the other levels. This is followed by a discussion of how and what type of contributions are made on the individual levels.

The idea for the SLM comes from the OSI (Open System Interconnection) reference model in the field of computer networks [77, pp. 28].

Physical The emitter and receiver components of a sensor are the main parts of the physical level. That is, the components that performs the actual emission and reception of signals in whatever form the current sensor employs. For an acoustic sensor this level includes the loudspeaker and the microphone. For an optical sensor the level includes the light source and the photosensitive receiver.

Data link The main purpose of the data link level is to provide a suitable interface between the processor level and the physical level. This interface has its own dedicated level since the signal processing, the emitter, and the receiver often are of quite different nature, and the interface thus becomes an important part of a sensor. The data link level includes the handling of blocks of samples often used in digitalized sensors. This means the timing of emitting and receiving individual samples, and storing the received samples for processing. This part of the data link level is often handled by the signal processing hardware.

Processor The hardware that constitutes the platform for the remaining levels are included in the processor level. The content on this level ranges from an oscillator and a comparator in very simple sensors to any type of signal processing hardware such as microprocessor, DSP, FPGA, etc.

Signal All the signal processing dedicated to providing the signal with given mathematical properties and to extracting information about the transmission channel belongs to the signal level. The description of the operations performed at this level is in the form of algorithms and software dedicated to connect this level to the processor level. The tasks performed at this level often require the main part of the resources provided at the processor level.

Control The variable parameters in the signal processing is controlled at this level. The purpose of the control level is to optimize the signal algorithm in given situations. This level is often closely related to the signal level. The control level is responsible for handling well-defined, short term, temporary changes.

Decision The performance at the control level is evaluated at the decision level to determine if some kind of change of the mode of operation is needed, for instance in response to significantly changed SNR or failure in some part of the sensor. The decision level typically handles long term changes of the sensor operating conditions, and thus responds relatively slowly compared to the control level.

Functionality The task of all the previous levels combined is obtaining information about the environment of the sensor as accurately as possible. The functionality level uses this information to determine the proper output of the sensor, and thus essentially provides the functionality of the sensor. In simple sensors this level might use the comparator at the processor level together with a threshold or saturation to produce an on/off output signal. In more advanced sensors such as the 3D sensor of Part II of the thesis the functionality level consists of a series of complicated mathematical algorithms, and takes up a significant part of the resources provided at the processor level.

Interface The input and output of the sensor is handled at the interface level. Often the output is an electrical signal to some external control mechanism. The input is typically potentiometers, buttons, and the like, by which the user can change certain parameters of the sensors functionality.

3.1.3 Contributions at Different Levels

In the sensor industry most of the effort for increasing the efficiency of sensors is in the three lowermost levels, in particular the physical level. The major manufacturers of emitter and receiver components, such as Siemens and Texas Instruments to mentioned a few (there are very many others), constantly present new components with increased efficiency, smaller size, quicker response and so on. The number of new components is so large that many manufacturers have quarterly magazines for presenting these. A merging of the physical and data link levels is sometimes seen as integrated circuits that combines

the receiver and the electronics, and occasionally also the emitter.

As mentioned previously the signal processing technology has undergone an extensive development in the past years. The process is still going on, and we can expect to see increased performance in many years to come. This development is by no means spearheaded by the sensor industry, but the industry will certainly benefit from the development, nonetheless.

The four levels from signal to functionality has received far less attention then the first three. This is in no small part because it is possible to construct a sensor with very limited content on these levels (and indeed many sensors are constructed that way), whereas the concept of a sensor without the three first levels does not make sense (sic!). The content at the higher levels is more descriptive of nature, and often presented in the form of algorithms. While most sensor manufacturers acquire many of the electrical components from sub-contractors, it is not customary for sensor manufacturers to acquire 'mathematical' components from external parties. Consequently, the development of these components happens to a large extent in close environments rather than across an entire industry.

This thesis provides suggestions and ideas for algorithms in the signal and functionality levels. The challenge of increasing the robustness undertaken in this Part I also includes some means for handling the tasks at the control and decision levels. The 3D sensor presented in Part II is contribution to the functionality level only.

3.1.4 Sensor Principles

The exists a wide choice of sensors in the market. The applications areas for sensors are numerous and the diversity of sensors within each area is big. Nonetheless, the majority of sensors can be classified into a few well-defined categories of basic sensing principles. Such a classification is given in Table 3.1. Note that this table shows a rather coarse classification and that the list of applications serves only as a guideline as to what the sensors are typically used for. The table also for comparison contains the most common passive sensing principles.

Table 3.1: Sensor principles with typical fields of applications.

Active		Passive	
Principle	Applications	Principle	Applications
Optical	Proximity, position	Optical	Proximity
Acoustic	Distance, flow	Temperature	Temperature
Inductive/capacitive	Proximity, level	Electrical	Force, load, pressure
Magnetic	Position	Mechanical	Proximity
Microwave	Movement, position	Chemical	Concentration

The distinction in sensing principles is important when discussing the signal processing part of a sensor in relation to the parameters listed in Section 3.5. Some properties are more easily provided with one principle compared to another principle. For instance an optical proximity sensor typically has a much smaller response time than an acous-

tic sensor simply because of the difference in transmission speed of electromagnetic and acoustic waves. Incidentally, for the same reason an accurate distance sensor is more easily deviced using the acoustic sensing principle.

3.2 Applications of Active Sensors

Sensors and sensor systems perform a diversity of sensing functions allowing the acquisition, capture, communication, processing, and distribution of information about the states of physical systems. This may be chemical composition, texture and morphology, large-scale structure, position, movement, pressure and load, flow, etc. Sensors can act as the link between an actuator and a decision process, and as a mean for recording or visualizing the state of a physical process.

While sensors are used in very many different applications the individual sensor is rarely capable of performing anything but a highly specific task. It is a characteristic feature of a sensor that the device is tailored to the environment in which it is to operate. And these environments are indeed quite different. A list of applications areas of sensor products from Banner (a major manufacturer of sensors) is given in Table 3.2. The list

Table 3.2: List of industries using Banner products.

Air conditioning	Grain processing	Packaging
Aircraft & aviation	Hazardous areas	Paper manufacturing
Agriculture	Heating	Pharmaceuticals
Assembly	Industrial machinery	Plastics manufacturing
Automated storage	Inspection	Power transmission
Automotives	Iron manufacturing	Printing industries
Computers	Material handling	Raw materials processing
Converting	Measurement	Robotics
Conveyor control	Medical manufacturing	Semiconductor manufac.
Dairy processing	Metalworking	Systems integration
Electronic equipment	Microelectronics	Textiles
Factory automation	Mining	Transportation
Food and beverage	Motion control	Wastewater treatment
Forest industries	Shipping & handling	Wood processing

Source: Banner web site

includes many different areas, and each item in the list covers a long series of applications. While this list is by no means exhaustive it gives an impression of the extent of the sensor market. It is clear that sensors in general have a very broad range of applications and that there is a sound basis for further development of sensors.

3.2.1 The Sensor Market

Sensor technology is one of the technologies that will play a major role in the future. The current world market for sensors is estimated at 150 billion EUR with an annual growth of around 15 per cent. This means that market growth for sensors is considerably higher than

for industry in general, but lower than what has been observed in information technology. This sensor market is divided approximately evenly between passive and active sensors. Two of the most interesting sensor products of the next decade with respect to market volume is optical sensors and multiple sensor systems.

There has been a trend in the sensor market for centralizing the production which means decentralization of measurements. This leads to a shift in focus from final product control to process control or even sensing of raw materials. This typically requires smaller and more reliable and robust sensors. The same requirements are also supported by the movement away from invasive towards non-invasive or non-contact sensors.

Sensors are used in practically all sectors of industry. They may be essential for a given product or process or they may provide the value-added that makes the process or product competitive. Knowledge about sensors, their applications, and their future developments thereby helps to position companies and research institutes to grasp emerging opportunities.

All the information in this subsection is from the Sensor Foresight Report [73].

3.3 Existing Sensor Implementations

The huge variety of sensors in the market means that there exists very many different ways of implementing the various parts of an active sensor. In this context it is especially interesting to learn the most common types as well as the more recent types of implementations of the Signal generator and the Signal processing parts, see Fig. 3.1. However, it is no simple task to gather this information, partly because of the diversity (and secrecy) in the sensor industry, partly because research results in this area are published in a variety of literature. While the author during the past few years have gathered some knowledge on the traditional sensor principles (see below), it has been difficult to find any references to previous research on signal processing solutions for low-cost active sensors. In fact, all references to development of low-cost sensors that the author has been able to find is focused, one way of the other, on reducing the cost or improving the performance of the hardware. This does not mean that signal processing is not a research area in the field of sensors, but to the best of the author knowledge the research is focused on providing increased functionality by means of complex algorithms rather than providing robustness and reduced cost by means of simple and efficient algorithms. This is especially so for sensor-dedicated publications such as IEEE's Sensors Journal, Wiley's Sensors Update, and the internet-based Sensors Online.

The consequence of this is that the first part of the thesis is not very well supported by references to previous research results, at least not with respect to the structure and overall choice of methods. The individual signal processing methods, in particular those based on mathematical considerations, are in most cases supported by references.

A majority of the sensors in the market employ traditional frequency-based detection methods. This basically means that harmonic signals in on form or another is used. This is not only the case for analog sensors, but often also for sensors employing digital

hardware. The author believes that the reason for this is that the basic skill needed for designing sensors is electrical engineering, and the primary mathematical tool in this field is frequency analysis in the form the Fourier or Laplace transforms, transfers functions, filter theory and the like. With such concepts in mind multiplexing in the frequency domain is an obvious solution to the challenge of designing signals for active sensors. The signal itself can be a sinusoid, a square wave, a sequence of repeated pulses, and other frequency (and time) localized forms.

Admittedly, a frequency-based approach is in many respects and in many cases a reasonable choice, and the author acknowledges the fact that most sensors do function correctly in whatever application they are being used. However, this thesis is a response to the fact that many sensors are sensitive to frequency-localized noise, and the author is aware of several examples of sensor products failing to behave appropriately in environments which should not have posed a problem.

3.4 BeoSound Ouverture

The Ph.D. study which has led to this thesis was initiated as a response to a desire of Bang & Olufsen to develop a new type of sensor for some of their products. In particular, the CD player BeoSound Ouverture employs an infrared sensor to detect the presence of a hand. Two glass doors covering the front slide aside when the user wants to operate the keyboard and the CD drive. The infrared sensors are locate behind two panels which appear black, but they are transparent at near infrared wavelength. The Ouverture is shown in Fig. 3.3. The detection system in the CD player is completely analog and is

Figure 3.3: The sliding doors in the BeoSound Ouverture is activated by the presence of a hand. The hand is detected by infrared sensors behind the black panels in each side of the CD player. Source: B&O web site.

based on a high amplification in the optical feedback loop combined with saturation to indicate detection. The systems has a very low latency and thus gives the impression of reacting instantly to an approaching hand.

Designing the detection system is a surprisingly difficult engineering challenge. The specifications describes a system with almost zero standby power consumption, low response time, robust to all types of typical and less typical optical and electrical disturbances in a domestic environment, requires no maintenance, and capable of withstanding many years of wear and tear without failing. While the detection system does work according to specifications it did require a significant amount of resources to get that far.

It is therefore obvious to ask whether using a digital solution, as suggested in this thesis, could reduce time and cost of developing a similar system in the future.

The two major challenges in this case is the very limited computational power available and the desire to have a method which can handle all types of noise that will emerge during the next many years. This calls for a detection algorithm which is capable of adapting to changing noise conditions without the need for a thorough analysis of the noise. At the same time the cost constraint limits the available processor to being a low bit resolution fixed point processor. The detection method therefore has to be numerically very stable, too.

3.5 Sensor Performance Parameters

The range of applications for sensors is very wide, and the number of different sensors is huge indeed. Since this thesis aims at providing better means for active sensing altogether, it is necessary with some mean for comparing such a variety of sensors. To do this it is convenient to have a series of parameters which are independent of field of application and sensor principle. Consequently, it is necessary in this context to have a more abstract view on sensor parameters than provided by data sheet and the like. For that purpose this section lists a series of such parameters. It should be noted that this list is compiled to suit this thesis, that is relatively low-cost sensors (see the low-cost property) for non-extreme conditions. Thus, it does not include some of the parameters which might be relevant for high-cost sensors such as radar equipment, highly reliable medical equipment, high-precision laboratory equipment, and sensors for other extreme conditions such as high temperature, high speed, and microscopic size.

The parameters in the following list are generic in the sense that they apply to all types of active sensors, independently of sensing principle. At the same time these parameters are closely related to the sensing principle employed. Therefore, this list is a good tool when comparing methods of active sensing. This applies to traditional as well as new methods, the latter being the main purpose of this thesis.

Robustness The robustness of a sensor describes the ability of the sensor to generate the correct output in given situations. For instance, in the case of a proximity sensor a high robustness means that the sensor is able to consistently determine the presence of an object without responding positively whenever there is no object to detect. The degree of robustness determines to what extent severe and different types of disturbances can be handled properly. The robustness thus describes how well the sensor responds to noise which is significantly more powerful or of a different sort than the noise (conditions) for which the sensor was design to operate under. Robustness does not necessarily mean that the sensor is capable of maintaining the same precision or response time in all conditions, but that the sensor can recognize disturbances, and handle accordingly. That is, the sensor still behaves predictably instead of giving a random or saturated output.

Immunity The ability of the sensor to ignore external disturbances is called immunity. Such disturbances are often a priori unknown except for some general characteristics (unlike internal disturbances which are usually of a well-known nature). They are in many cases generated by new products (ranging from cellular phones and energy saving light bulbs to production machines and measuring equipment) emitting previously unseen electromagnetic or acoustic signals. Note that immunity is a necessary, but not sufficient condition for robustness.

Adaptability The ability of the sensor to adapt to changes in the environment. This includes fast changes such as electromagnetic signals from cellular phones, slow changes such as lighting conditions, and long term changes such as aging. The adaptability of the sensor can be one of the important factors in making the sensor robust.

Response time The time elapsed from the occurrence of a detectable state to the response of the sensor is called response time. This varies dramatically depending on the applications. The fastest sensors interesting in the context of this thesis (small proximity sensors) has a response time in the order of 10^{-5} s, while the slowest (level reading in tanks and containers) has a response time in the order of 10^{2} s. The requirements for response time typically affects the accuracy and cost of the sensor.

Accuracy The accuracy of a sensor is the quantification of the possible difference between the true and the reported value of the sensor variable. Depending on the context in which the term 'accuracy' is used, this can be a physical variable such as angle or distance, it can be an electrical variable such as received intensity, and it can be a signal processing variable. In case of the latter the accuracy often refers to the precision of internal computations, especially in fixed point and dedicated hardware. The accuracy on all levels are primarily governed by cost and response time.

Flexibility The ease with which a sensor can be reconfigured is called flexibility. Such reconfigurations range from simple updates for correcting minor problems to new products based on old ones. The difference between versatility and flexibility is that the former applies to final products, whereas the latter applies to the product platform.

Versatility The number of different uses for a given sensor determines the versatility. This property is clearly interesting from the consumers point-of-view. For the manufacturer high versatility can be a two-edged sword: A sensor which can be used in many different applications is produced in larger quantities (typically increasing the contribution marginal), but slight changes in consumer demand does not render the product useless (thus, there is no need for more, new products).

Reliability The sensors ability to withstand wear and tear as well as extreme physical conditions is called reliability. A sensor which functions correctly for many years despite of dust, scratches, component aging, corrosion and the like is considered reliable.

Intelligence The ability of the sensor to make decisions based on reason and available information, and the ability to handle unforeseen situation is called intelligence. Depending on the degree of intelligence this ability introduces to some extent robustness, adaptability, flexibility, and versatility simultaneously.

Low-cost It is obviously desirable to have production costs as low as possible. In the case of mass-produced sensors it is particularly desirable to keep the variable costs at a minimum. The sensors considered in this thesis are of the cheaper kind (typically 5 to 25 EUR in production cost for a complete sensor component).

Reduced size The size of the entire sensor is an important factor in many applications. A sensor is usually a component in a larger product, and less often a stand-alone product. Consequently, a sensor must fit into whatever frame or base is available. In general, this calls for the sensors to be small. Note, however, that for the sensors considered in this thesis the size requirement is always secondary to the cost requirement.

Fault tolerance The fault tolerance refers to ability of the sensor to detect and handle mechanical and electrical malfunctions. Fault tolerance contributes to the robustness in the sense that the ability of the sensor to properly handle a fault ensures a correct output in an abnormal situation (see the definition of robustness).

Self-calibration Easy operations is a key issue in many sensor applications since the user often is not familiar with the sensor construction or the sensing principle. At the same time most sensors needs calibration. Self-calibration allows the user to install the sensor, and to change the operating conditions without worrying about subsequent calibration. This is therefore an important feature in many cases.

Table 3.3 gives an impression of approximately how the most common sensing principles rates within the scope of these parameters. The wide variety of sensors means that the ratings in this table is only intended as a guide. The 2nd and 3rd generation sensors might employ any of the sensing principles (and new ones, too), and the ratings are intended to express the (author's) expectations to these sensors.

Table 3.3: A coarse classification of the various sensor types.

	Robustness	Immunity	Adaptability	Response time	Accuracy	Flexibility	Versatility	Reliability	Intelligence	Low-cost	Reduced size	Fault tolerance	Self-calibration
Mechanical	●	●	–	●●	●	–	●	●	–	●●	●	–	–
Optical	●●	●●	●	⁂	●●	●	●●	⁂	–	●●	●●	–	●
Sonic	●●	●●	●	●	⁂	●	●	●●	–	●	●	–	●
Magnetic	●●	⁂	–	⁂	●	●	●	⁂	–	●●	●●	–	–
Capacitive/inductive	●	●●	–	●●	●	●	●●	●●	–	●	●	–	●
2nd generation	⁂	⁂	●●	●●	●●	●●	●	⁂	●	●	●	●	●●
3rd generation	⁂	⁂	⁂	●●	●●	⁂	●●	⁂	●●	●	●●	●●	⁂

● to a low degree, ●● to some degree, ⁂ to a high degree, – not at all.

There are two other parameters which are of interest in many types of sensor systems, but they do not fit well into the previous list. The first is the ability of a sensor to function

in, or even benefit from being in, an environment with other sensors of the same type. Not all types of sensors are sensitive to the presence of other sensors. Usually the sensors with a large range will interfere with each other unless precautions are taken. Optical and sonic sensors are examples of sensing principles which are likely to disturb other similar sensor systems. As long as the sensors use different modulations and carrier frequencies the problem is minimal, but two sensors of exactly the same type and model might cause problems.

Larger systems that employ sensors usually have several identical sensors located near each other. For instance, proximity sensors along a conveyor belt or an assembly line. It is obviously important that they do not disturb or interfere with each other. Since the sensors use the same signals the individual sensor must also be able to handle that other sensors operate simultaneously. This is complicated by the fact that most low-cost sensors are autonomous units with slightly varying performance specifications between units. In particular, the frequencies used for signal emission and reception is usually not exactly the same for any two sensor units.

The other property is synchronization between emitter and receiver. In some cases the sensor consists of two physically separated units containing an emitter and a receiver, respectively (i.e. through beam sensor). It then becomes necessary to have some sort of synchronization to assure that the receiver records the correct transmission signal. This can be done externally, i.e. by some other form of communication, like a RF link. Alternatively, the receiver can try to synchronize on the emitted signal directly. If the signal processing algorithm in the receiver is capable of synchronizing on the received signals this solution is often preferable due to the lower cost (this only requires more processing power in the receiver). While a particular transmission sequence might be optimal with respect to the noise conditions, it might not the useful for synchronization. Thus, it possess an additional challenge to design transmission signals which are optimal in both respects. While the author has indeed investigated various ideas for synchronization, the work is not reported in this thesis.

If the emitter needs to transmit information to the receiver this can often be done by means of the two sensor units themselves, since there is typically ‘connection’ between the two units most of the time. Optimally, the information should be modulated in the same way as the ‘sensor signal’, because then the transmitted information signal can be used for sensing, and consequently, there is no time gap in the sensor output. Transmitting information from receiver to emitter needs some extra means of communication.

Methods for Measurement of Channel Gain

4

The basic concept that governs the choice of methods and means for making channel gain measurements in this thesis is improvement on a number of parameters in active sensors. The most important parameter is robustness. The background for this was discussed in the previous chapter. The main tool for achieving the improvement is signal processing algorithms, and in this chapter the focus is on designing an algorithm that fulfills a series of performance requirements. The structure of the algorithm is based on engineering and applicational considerations while the individual steps in the algorithm to a large extent is based on mathematics. The algorithm is described in detail in this chapter, and applied to real signals in the next chapter.

4.1 Two Methods for Measuring Channel Gain

The main purpose of the channel gain algorithm presented in this part of the thesis is to estimate the change in intensity of a signal transmitted through a channel. For the algorithm to be interesting in the context of low-cost sensors it has to conform with a series of performance requirements. The performance parameters were listed in Section 3.5 in the previous chapter, and in this section they are quantified for the purpose of specifying the performance requirements of the algorithm. The generic algorithm has a variety of embodiments, and two somewhat different embodiments are presented in details in this section.

Ideally, the presentation starts with the performance requirements followed by the generic algorithm and ending with the two embodiments. However, the author believes that having the subjects introduced in this order will lead to a rather wearisome presentation. Instead, the two embodiments are presented first followed by the generic algorithm in Section 4.2. A discussion of performance requirements in relation to the algorithm, as well as to the two embodiments, is given in Section 4.3.

It is easier to understand the reasons for the chosen structure of the algorithm with two specific cases in mind, and it is also easier to see what performance requirements can reasonably be expected to be fulfilled with the generic structure in mind.

The following two sections describe two methods for determining the channel gain in systems with multiple emitters and receivers. They are both designed such that is it

easy to choose the trade-off between robustness and detection time by adjusting a single parameter. The first method is based on spread spectrum (SS) signals while the second method is based on wavelet generated signals. The first method is mainly useful in sensors where only very little computational power is available, in sensors where the detection must be fast, and in sensors where the noise is expected to be rather varying in nature. The second method requires somewhat more computational power and will often be a little slower than the first method. However, it is capable of in real-time adjusting to many types of non-Gaussian noise.

It is assumed in this chapter that the reader is familiar with the Rudin-Shapiro transform (RST) as well as the wavelet packet transform (WPT). A thorough description of the RST is found in Chapter 11 (which can be read independently of Part I and II). The WPT is a well-known transform, and there exists an incredible amount of literature on the subject. An easy introduction to the subject can be found in Jensen and la Cour-Harbo [45], a somewhat more extensive introduction is Wickerhauser [83], and a good reference for a purely mathematical presentation of wavelets is Daubechies [26].

4.1.1 Using Spread Spectrum Modulation

The basic idea of the SS based algorithm is to combine SS modulation with a multiplicity of transmission channels, i.e. the channels are separated in the 'spread spectrum domain' or code domain (just like radio broadcasting is a separation in the frequency domain).

To perform the sensing one or more emitters simultaneously emits a single ping, like a sonar. For each emitter this ping is actually a short, typically 16, 32, or 64 samples long, SS sequence. Each emitter has its own sequence, and all the sequences are different from each other and each represents a channel in the SS domain. Each receiver then receives a mix of all the signals, which is processed to determine if there has been an occurrence (based on the intensity of the ping in each channel) to which the sensor should respond. The method reported in this subsection applies to the receivers individually, so in the following only one receiver is considered.

Since the channels are not separated in neither the time nor the frequency domain any noise occurrences localized in time or frequency will affect all channels more or less evenly. The idea is then to use only a few channels for the actual transmission while the remaining channels are used for detecting noise. That is, no ping is emitted into these channels. The algorithm is thereby capable of detecting when the noise is such that the transmission is corrupted beyond recognition. By changing the length of the short signals it is possible to easily change the trade-off between response time and robustness.

One of the key elements in the algorithm is the modulation method. Here it is proposed to use the Rudin-Shapiro transform (RST) for this purpose, since it has a series of useful properties (see Chapter 11).

The sensing starts with a set of very simple digital signals to indicate which channels are used for pinging. The signals are zero sequences with a 1 at the location of the chosen channel. These signals are called designed signals. At the top of Fig. 4.1 three such signals

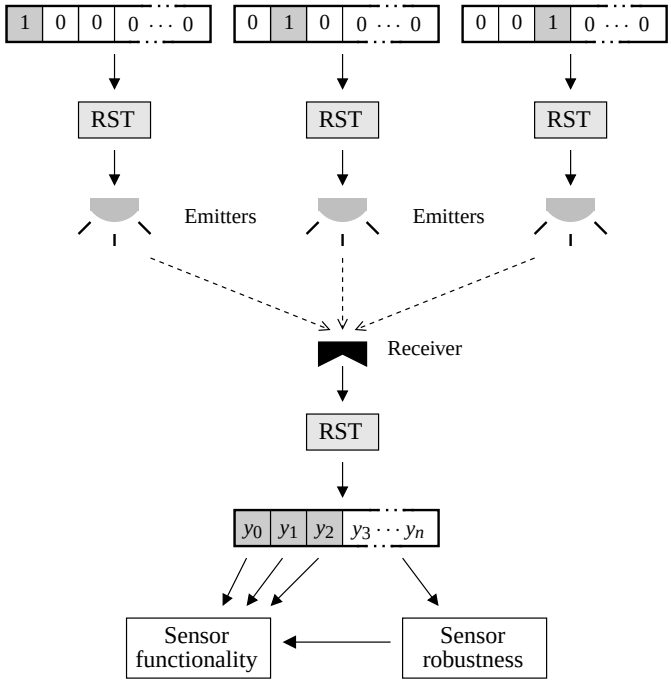


Figure 4.1: The Rudin-Shapiro transform is used to multiplex in the code domain. Here three channels are used, one for each emitter, and the remaining channels are used to increase the robustness. Note that this figure does not include the electronics and the other necessary signal processing algorithms.

are shown. An RST of such an almost vanishing signal produces a binary SS signal with a number of samples equal to the original, designed signal. The transformed signals are transmitted by a number of emitters to the receiver through the sensor environment, and the received signal is now demodulated with the RST. If the receiver behaves in a linear fashion then, because of the orthogonality property of the RST, the entries in the demodulated signal corresponding to the chosen transmission channels now holds the energy of the received pings (y_0 holds the energy from the first emitter, y_1 holds the energy from the second emitter, and so on) while the remaining entries y_3 through y_n are zero.

In any real life application there is obviously noise present, and thus the remaining entries are not zero. However, in an ideal sensor they always remain unaffected by the transmission from the emitters, and can therefore be used to determine properties of the current noise. This information can in turn be used to validate the quantities obtained from the transmission from the emitters. In a real sensor some inter-channel cross talk is to be expected, however, if for instance the emitters and receivers use the same power supply or the transfer function for the channel or the electrical circuit is not constant in the used frequency range.

Two validation methods are reported later in this chapter in Section 4.9.

4.1.2 Using Wavelet Modulation

The spread spectrum modulation is useful in scenarios where the noise is far from stationary or where only little computational power is available. But in a scenario where the noise is close to being stationary the SS method's lack of ability to adjust to detectable time and frequency-localized noise makes it less effective than a joint time-frequency (JTF) modulation. This is not surprising given that the purpose of SS modulation is to distribute energy evenly in time and frequency. By using the wavelet transform for modulating signals it is possible to achieve a high degree of control of the distribution of energy in time and frequency. Consequently, it is also possible to adapt a transmission signal to noise occurrences which are stationary in the JTF domain. For more information on JTF analysis in general, see Qian and Chen [64].

The sensing in the wavelet modulation case is performed in much the same way as in the RS modulation case: One or more emitters simultaneously emits a single ping. For each emitter this ping is a wavelet modulated sequence, and each emitter has its own sequence. Each receiver then receives a mix of all the signals, and the received signal is processed to determine if there has been an occurrence to which the sensor should respond. As in the RS case the wavelet method applies to the receivers individually, so in the following only one receiver is considered.

The signal generating process is a little more complicated than in the RS case (which is indeed very simple). First a number of signals are designed. Each of these signals is vanishing except on some interval which differs from signal to signal. Each interval corresponds to an element (or sub-band in the frequency interpretation) on some level in a

wavelet packet decomposition, i.e. it is on the form $[m \cdot 2^{J-j}; (m+1) \cdot 2^{J-j} - 1]$ for the m'th element (or sub-band) on the j'th level (m and j counts from 0) in a WP decomposition of a signal of length 2^J. Three such signals are shown on the top in Fig. 4.2. These

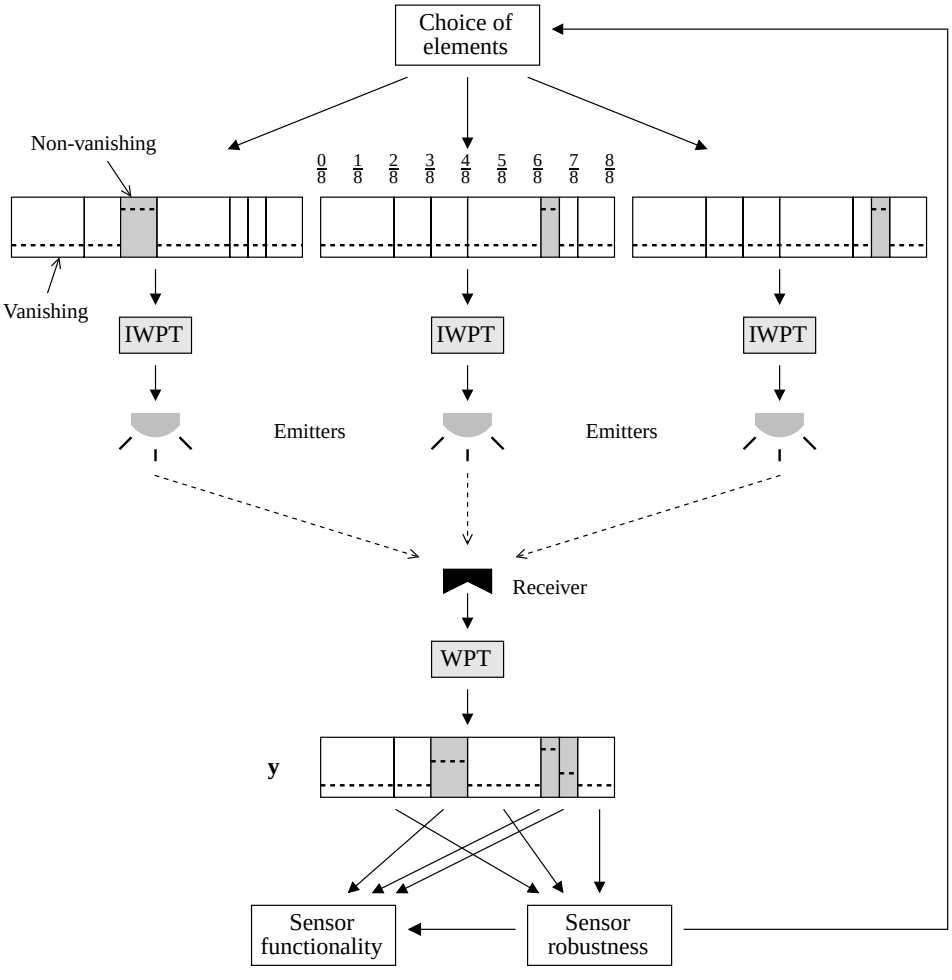


Figure 4.2: The wavelet packet transform is used to multiplex in the joint time-frequency domain.

signals are non-vanishing on intervals corresponding to the 3rd element on the 3rd level, and to the 12th and 13th elements on the 4th level, respectively. Note that the signals are typically not constant on the non-vanishing part since this yield transmission signals with a rather narrow frequency content. Ideally the zeroth moment of each designed signal is zero (see Section 4.6.2).

Now the inverse WPT (IWPT) is applied to each signal. The basis chosen for this transformation must comply with the chosen non-vanishing intervals. In Fig. 4.2 the vertical lines in the signal show what basis has been chosen in this particular case. In the noiseless case the received signal becomes a mix of the transmitted signals, and a subsequent forward WPT will produce a signal which is a linear combination of the original, designed signals. To determine the intensity in each channel the inner product is taken between the received, transformed signal and each of the designed, original signal. This gives a number which is a relative measure of the transmitted intensity.

In Fig. 4.2 there are three emitters and thus three intensities are measured. However, the designed signals might easily have a length, say N, such that there are more than a total of three samples in non-vanishing parts in the received, transformed signal. Note that there has to be at least $N = 16$ samples for the WPT to make sense in the Fig. 4.2, and in this case the non-vanishing parts have 4 sample in total. Since there are more non-vanishing samples in the received, transformed signal than intensity numbers coming out of the measurement, there is a potential for using the remaining channels for determining noise characteristics, just as in the case of the RST method discussed in the previous subsection. But where the SS channels corresponded to single samples in the $\mathbf{y}$ signal, see Fig. 4.1, it is a bit more complicated with the wavelet modulation.

If the number of samples in the signals in Fig. 4.2 is $N = 256$ then the number of non-vanishing samples in the first designed signal is $256/8 = 32$, and the second and third has $256/16 = 16$ non-vanishing samples each. Thus, the received, transformed signal $\mathbf{y}$ has a total of 64 non-vanishing samples in a noiseless transmission. Since the inner products yields only three values, there is room for 61 estimates of the transmission noise. These can be obtained in the following way. The first transmission signal consists of 32 non-vanishing samples. Now, let $\mathbf{u}_0$ through $\mathbf{u}_{31}$ be 32 orthogonal vectors in $\mathbb{R}^{32}$, where $\mathbf{u}_0$ is the original, designed signal. Inner product between $\mathbf{y}$ and the 32 $\mathbf{u}_k$'s will give a signal with the same property as the $\mathbf{y}$ signal has in the RS method, i.e. a signal with the first sample being the channel gain and the remaining samples being an indication of the noise level. The same procedure is applied to the two other transmission signals (although only 16 $\mathbf{u}$' s are needed for these signals).

Of course, there is also the parts of $\mathbf{y}$ which is pure noise, and thus can be used right away for determining noise characteristics.

Just as the RST the WPT responds in a easily predicted fashion to time and frequency-localized noise occurrences. But in contrast to the RST it is fairly easy with the WPT to reduce the effect of such localized occurrences. This is basically because the WPT is a JTF modulation. Since each element in the WP decomposition represents the whole time line of the signal, a time-localized occurrence remains localized in each element after transformation. Consequently, it makes sense to apply various methods after transformation for removing transients from a signal. At the same time the WP decomposition yields a band pass filtering of the signal and thus frequency-localized noise will only shown up in one or two elements. Assuming that the noise is stationary it is easy to determine on the basis of the received signal which elements in the WP decomposition will have the lowest

noise energy in subsequent transmissions. This information can then be used to alter the designed signals which determine the frequency bands used for transmission. The details of these methods are discussed in Section 4.4 and 4.7.

4.2 Suggested Algorithm

The two methods for determining channel gain described in the previous two subsections are special cases of a more general algorithm. In this section this general algorithm is introduced and discussed. The general algorithm is presented after the two special cases because, as mentioned in the beginning of this chapter, the author believes that having the two embodiments in mind will make it easier to see which steps are necessary and to imagine what the content and importance is of each step in given scenarios.

There is a long series of relevant requirements which ought to be taken into account when designing the algorithm. These requirements are based on the various sensor performance parameters listed in Section 3.5. The quantification of each requirement obviously depends on the application, but as described in Chapter 3 some of the performance parameters are in general interesting in the context of this thesis. Especially robustness, response time, and low-cost are essential parameters. Accordingly, the signal processing steps of the algorithm provide a good performance with respect to these parameters.

4.2.1 Basic Idea

The suggested algorithm is basically divided into six parts, which are two hardware parts, three signal processor parts, and one external part. The contribution from this thesis is almost entirely in the two signal processor parts. The other parts have been included in the following description of the generic algorithm to provide a proper context. A schematic presentation of the algorithm is shown in Fig. 4.3.

The basic idea is to start with a set of simple, designed, digital signals which are transformed by an invertible transform to create the transmission signals needed in the active sensor. The signals are converted from digital to analog format, and then emitted and received. After proper analog processing the received signals are converted to digital form. The post-processing consists of inverse transformation, various types of denoising, and estimation of transmission conditions. The latter includes the purpose of the entire process, namely to determine the channel gain, i.e. the transmitted intensity.

The steps which the author believes are necessary for a proper implementation of this idea are shown in Fig. 4.3, and discussed individually in Section 4.2.2 and 4.2.3. Each of the signal processor steps covers quite a lot functionality. There are many different signal processing methods which can be used to obtain this functionality. The more interesting methods are discussed in further details in the remaining sections of this chapter. Appropriate references to these sections are given in the description list in the next subsection.

Note that although the descriptions of the algorithm throughout this and the following chapter are fairly detailed and constructive, it is still necessary, in a given embodiment of

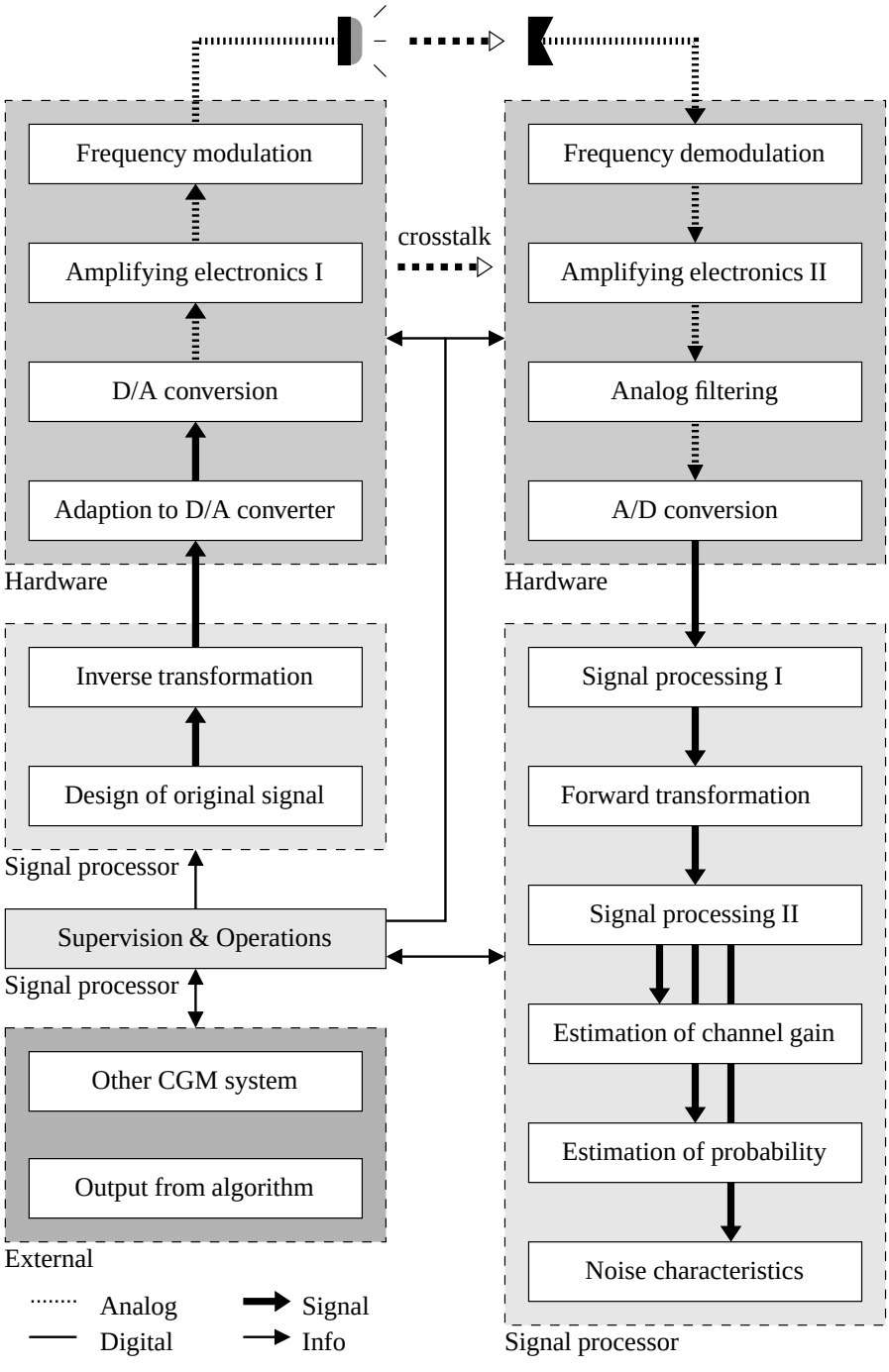


Figure 4.3: A schematic view of the generic channel gain algorithm.

the algorithm, to carefully design, test, and tweak each step to accommodate exactly the desired functionality.

4.2.2 Signal Processing Steps in the Algorithm

Each of the steps in the algorithm has a specific purpose. In this and the following subsection the steps are presented one by one and the key elements in each are discussed. Since the focus in this thesis is on the signal processing part these steps received the most attention, and are presented first. Following this in the next subsection the hardware steps are presented along with the external communication.

Design of original signal There are basically two concepts when designing the original signal. Either the design aims at spreading the energy in whatever domain the noise is localized, in order to reduce the impact of the noise, or the design aims at localizing the energy in the noise domain, but at different locations than where the noise occurs. The former method will have a low, but usually more or less constant sensitivity to localized noise, whereas the latter method will have a (potentially) very low sensitivity to localized noise.

The spreading method is useful when the localized noise has a highly unpredictable behaviour, and since this method by its very nature does not require adaption it is easier to use and cheaper in terms of computational power. It also means that the design of the original signals becomes very easy since all the work in creating the spread spectrum signal is done by the subsequent transformation.

The localizing method is useful when the noise is stationary in the localized domain. In this case the original signal is a simple, though not trivial, signal which is designed in accordance with the various information gathered during past transmissions. In particular, the design is based on the noise characteristics combined with the properties of the inverse transform to construct a signal which is close to orthogonal to the expected noise. This property guarantees a good separation in the localized domain.

The concept of designed signal are discussed in more details in Section 4.5.1.

Inverse transformation This transformation will convert the designed signal into the transmission signal. The transmission signal will then have properties which, hopefully, will allow the post-processing to do proper denoising, estimation of channel gain etc. The entire transform might be fixed, or it might have a set of parameters which allows for some degree of adaptation. For instance, the RST is a fixed transform, while the WPT can use the best basis algorithm to search for the best location in the WP domain to place the transmission. See Fig. 4.2 where a particular basis was chosen, and Section 4.5.5 on finding holes in the noise.

While the only mathematical restriction on the transform is that it is invertible, there are in many cases rather stringent restriction on the numerical stability of the transform, and often it is necessary that the transform has a faster than $O(N^2)$ implementation, as is standard for linear transforms, because otherwise the real-time

requirement can only be fulfilled with a rather powerful signal processor, see Section 4.3.3.

Note that this step is called Inverse transformation because in the case of the WPT this is the usual way of naming the transform which maps a number of frequency bands into one 'all-frequency' signal. For some transforms the order of transformation does not matter at all, like the symmetric RST. For the set of transforms which are only invertible from one side, i.e. the transform matrix is not square, it is of course vital to have the right order of transformation. The Gabor transform is an example of this phenomenon.

Forward transform After transmission when the signal is back on digital form it is subjected to a forward transform (i.e. inverse of the transform applied prior to transmission). For a noiseless transmission this will reconstruct the original, designed signal since the transform is invertible. However, since there are indeed always noise it is necessary to include a number of 'noise-handling' methods in the post-processing. This starts with the transform itself.

The main part of the noise energy is typically acquired at the receiver. Since many types of receivers behaves in a close to linear fashion it is often reasonable to assume that the noise is additive. Consequently, the signal can be 'separated' from the noise by a linear transform, i.e. transformation reconstructs the original, designed signal with additive noise. If, however, the noise is convolved with the signal a deconvolution is needed. Such a procedure is in most respects much more difficult than a linear transform, and it is outside the scope of this thesis to investigate the effects of convolution noise.

Beside linearity it is also important that the transform maps small perturbations into small perturbations. This ensures that a small noise contribution has a small impact on the transformed signal. A linear transform might lack this property if a number of the basis vectors are close to being linearly dependent, i.e. the angles between the vectors are small. By using a orthogonal transform this problem does not exist, since energy is preserved under orthogonal transformation. It is not desirable to require the transform to be orthogonal at all costs, however. For instance, the classical 9-7 wavelet transform is not orthogonal, and some implementations of the wavelet transform on a finite interval are not orthogonal, either. The orthogonality aspect of the transforms is discussed in Section 4.5.4.

Signal processing I/II While the transform does the job of reconstructing the original signal it does not (necessarily) denoise the signal and prepare it for estimation of transmission conditions. Therefore two signal processing steps are including in the algorithm. Typically, the first step (I) does the main part of the denoising, while the second step (II) mainly prepares the signal for estimation of channel gain, noise level etc.

The denoising in signal processing I can consists of various methods for removing high energy noise occurrences prior to transformation. For instance, large transients can be removed, and low and high frequency noise can be reduced by filtering.

Though the chosen transform is suppose to handle such occurrences nicely, a sufficiently powerful noise burst will inevitably reduce the accuracy of the channel gain measurement. On the one hand it is therefore desirable to decrease the noise energy as much as possible prior to transformation. On the other hand there is a limited computational power available for denoising, and the applied methods must therefore be a trade-off between complexity and efficiency.

It is obviously desirable to apply denoising methods which handles the transmitted signal as gently as possible, or, alternatively, subject the transmitted signal to an easily predictable alteration (which then can be 'undone' prior to the estimation steps). This will make the result of the estimation steps more accurate. How to design the denoising to fulfill this desire depends to a large extent on the chosen transform, especially whether it is a spreading or localizing transform. A number of denoising methods adapted to the two previously suggested transforms are presented in Section 4.7.

After denoising and transformation the signal goes through the second signal processing step to prepare it for extraction of information. Here any alterations of the signal caused by the first signal processing step is handled. Any method used for this is obviously heavily dependent on the denoising and the transform. Then any other post-processing needed is applied. This could be more denoising such as transient removal (see Section 4.7.3), edge handling procedures, smoothing, etc. The outcome of this second signal processing step is a signal which resembles the original, designed signal as much as possible (except for the amplitude).

Estimation of channel gain The very purpose of the entire algorithm is to estimate the channel gain, so this step is obviously very important. It does not involve signal processing to the same extent as any of the other steps, however. The only task in this step is to acquire a value for the gain in each channel (i.e. from each emitter). These values (which will be denoted channel gain measurements, CGMs) are always obtained by inner products between the original signals and the received, transformed, denoised signal. The output is a number of CGMs, one for each emitter. The estimation is discussed in details in Section 4.6.

Estimation of probability To estimate just how accurate the CGMs are the Estimation of probability step evaluates the content in all the channels which were not used for transmission. Usually the number of available channels are much bigger than the number of emitters, and thus a pretty good estimate can be generated. The estimated accuracy can be used to determine whether the CGM is sufficiently accurate for further use, whether it is necessary to adjust the designed signal to adapt to changed noise conditions, and whether various parameters throughout the algorithm needs adjustment. An example of such an adjustment is the choice of basis in the WPT.

There is a lower limit to the accuracy of the CGMs which is determined by the white noise contribution (this limit can be lowered by filtering several consecutive CGMs, but this increases the response time). The white noise cannot be removed except by filtering, and it is therefore important to have a good estimate of the variance. This is best achieved by measuring the ℓ_2 norm of samples which are believed to be

unaffected by anything but the white noise. The validation methods presented later in this chapter both depend on a good estimate of the white noise variance. Some methods for doing this is discussed in Section 4.6.

The output from this step is partly an estimate of the accuracy which is for immediate use (validation of measurements), and partly information for adjusting algorithm parameters to the present noise conditions. This means that the output is used at the signal level and the control level in the sensor level model (see Section 3.1.2). A discussion of detection and validation is given in Section 4.9.

Noise characteristics It is not only the accuracy which can be determined based on the noise in the received signal. The noisy channels can also be used for detecting for instance if the sensor is experiencing some kind of failure or if the sensor is subjected to an overwhelming noise occurrences (one which saturates the sensor making denoising useless). This final step might for that purpose employ a series of methods for determining if the sensor has suffered from an extraordinary event. This thesis does not focus on this part of the algorithm, as it is strongly dependent of each particular application. The output from this step is used at the decision level in the sensor level model.

Supervision & Operations The entire process of measuring the channel gain is supervised and operated by this step. All decisions are taken here, and all the external communication are handled by this step. The Supervision & Operations may also control the various parameters in the hardware such as gain, analog filtering etc. This step comprises the control and decision levels, and in the case where the output from the algorithm is also the output from the sensor, this step also comprises the functionality level.

4.2.3 Hardware and External Communication in the Algorithm

The algorithm employs only standard hardware components in an effort to keep the variable expenses at a minimum. The following descriptions are therefore brief and mainly aims at visualizing the importance of each step in various scenarios.

Adaption to D/A converter The signal coming from inverse transformation is rarely suitable for D/A conversion. The signals needs to be scaled and shifted to fit the voltage range of the converter. This step will in some cases be handled by the signal processor.

D/A conversion Converting a digital signal to analog is a fairly straightforward process, and is usually handled by a separate electronic component. Note that when the signal is binary this step is not necessary. This is the case with the RS sequences. This is obviously an advantage, particularly in very low-cost sensors.

Amplifying electronics I The output from the DAC (or from the signal processor) needs to be fed to the emitter with just the right voltage and current. The amplifying electronics I, also called the emitter driver circuit, supplies the required power to drive the emitter. This step is almost always necessary as the DAC (or alternatively the

signal processor) rarely is able to supply sufficient energy for driving the emitter. An example of a driver circuit is given in Section 5.4.6.

Frequency modulation In some cases it is beneficial to abandon the base band in favor of some higher frequency band. This is highly application dependent and could be done by a standard modulator. None of the examples presented in this thesis uses a frequency modulation, though.

Frequency demodulation The modulated signal must be converted back to base band to suit the remaining steps.

Amplifying electronics II The Achilles' heel of the sensor hardware is the amplification of the received signal. Since the sensor usually by design is pushed to the limit the received signal is often very weak, and consequently a high gain is necessary to prepare the signal for A/D conversion. This means that the physical connection from receiver to amplifier is very sensitive, and a sloppy PCB or sensor design can therefore easily reduce the efficiency of the sensor.

Analog filtering While it is preferable to have an amplifier transfer function which is ideal for the transmission signals, it might in some cases be necessary to employ an extra filtering to reduce for instance high frequency noise. Filtering in the analog domain is a well-known technique and will not be discussed any further in this thesis.

A/D conversion Converting the signal back to the digital domain is essential for the remaining steps in measuring the channel gain. While the DAC is relatively simple the ADC is more challenging. Some microprocessors and most DSPs have a built-in ADC since the A/D conversion is an integral part of signal processing. The accuracy of the ADC (measured in bits) is an important information which determines a lower limit for what can possibly be achieved in the signal processing, and, ultimately, the sensitivity of the sensor.

Other CGM systems The presented algorithm can be part of a larger system where several CGM algorithms work together, possibly with other types of algorithms. By using multiple CGM systems it is possible to provide functionality which cannot be achieved with just one CGM algorithm. An example is the 3D sensor presented in Part II. Having several algorithms operating simultaneously also allows for exchange of information which can increase the efficiency of the individual algorithms.

Output from algorithm The ultimate purpose of the CGM algorithm is to provide an estimate of the channel gain and consequently this information is the primary output from the algorithm. The outcome of the validation procedure is in many cases also of interest, and may therefore be a part of the total output from the sensor. Sometimes other parameters, e.g. variance of the noise, the elements chosen in the WPT, are of interest also. The output is communicated to the functionality level in the sensor level model (see Section 3.1.2) or directly to the interface level in case the CGM output is the functionality.

4.3 Sensor Performance

The algorithms presented in the previous section are based partly on the potentials and limitations of low-cost sensor systems, partly on performance requirements for state of the art sensors. This sections presents and discusses the various aspects of sensor performance which forms the basis for the suggested algorithms in the previous section. First the physical constraints are presented in Section 4.3.1. This is followed in Section 4.3.2 by a discussion of how and to what extent the performance requirements can be achieved. Finally, in Section 4.3.3 the real time requirement is discussed in relation to the suggested algorithm for acquiring channel gain measurements.

4.3.1 Physical Constraints

The assumptions made about the physical framework are kept at a minimum to ensure a fairly general algorithm.

The signal The signal is an important component in an active sensor, and one of the key elements in the algorithm is the construction and post-processing of the signal. The physical embodiment of the signal depends on the application (though it is obviously electrical in the processing hardware). The emitted signal can be electromagnetic, acoustic, radioactivity, electrical, or a jet of water, for that matter. The algorithm does not depend on the form of the signal.

The channel The medium for transmission of the signal is called the channel. This is to be understood in a wide sense. The channel can be many different things such as air, water, wood, an electrical conductor, and so on. In some embodiments the channel also includes a reflecting or refracting object, like a hand or a prism. The algorithm does not include any a priori knowledge about the channel.

Number of signals It is assumed that an unspecified number of signals needs to be transmitted simultaneously through the same channel. The algorithm must allow for several emitters and several receivers to be operational at the same time, and it must be able to determine the channel gain for each combination of emitter and receiver simultaneously. It is assumed that all emitters and receivers are controlled by the same clock such as to synchronize emission and reception. It is also assumed either that the channel has sufficient bandwidth for an unaltered transmission, or that the channel transfer function is known. A non-constant transfer function typically causes inter-channel cross talk in the transform domain (e.g. spread spectrum domain or joint time-frequency domain), which to some extent can be countered by signal processing means.

Emitter It is assumed that the emitter and its driver circuit has the ability to convert the digital signal fairly accurately on whatever form the application calls for (acoustic, electromagnetic, etc.). This essentially means that response time and accuracy of the emitter is such that it is possible to generate the desired transmission signal, and that any non-linearity of the emitter is known. There are no assumptions on the size or

shape of the emitter, and neither does the algorithm presume any preset direction or location of the emitter.

Receiver The receiver must be capable of converting the transmitted signal into an electrical signal fairly accurately and without significant loss. Again this means sufficiently low response time, sufficiently high accuracy, and knowledge of any non-linear behavior in the conversion from optical to electrical power.

Amplifiers The amplifying electronics must have a transfer function which match the full range of frequencies in the signals. That is, the gain must be approximately the same for any frequency encountered in the signals. Alternatively, the transfer function for the amplifying electronics must be known.

Analog – digital conversion The accuracy of the ADC and DAC is not assumed to be known a priori. The conversion accuracy has to be fixed, though.

4.3.2 Accommodating the Performance Parameters

The list of performance parameters in Section 3.5 presented a number of parameters which are all relevant to address in most sensor applications. However, the suggested CGM algorithm is not suited for improving on all the listed parameters. As it has been hinted a number of times robustness is the primary concern of this thesis and the suggested algorithm has been designed accordingly. The two other important parameters are response time and the cost of the sensor. All three parameters are to some extent conflicting interests: Robustness can be obtained by waiting for more measurements, and by using better (and thus more expensive) electrical components, and the response time can be reduced by employing a faster ADC and faster signal processor. The algorithm is an attempt to design a method which allows for a trade-off between the parameters without the need for rethinking the entire process.

The algorithm does not only improve the robustness, but also some of the other listed parameters. The algorithm includes some tools for increasing the adaptability, for instance by the concept of finding holes in the noise by appropriate changes in the designed signal (see Section 4.5.5). This allows for short term adaption. However, adaptability is also about the ability of the sensor to switch between a number of modes depending on permanent changes in the operating conditions for the sensor. This subject is not addressed with the CGM algorithm.

The improved robustness does not automatically lead to increased accuracy. The accuracy parameter can often be increased at the expense of increased response time, but how to do this in an optimal way is not discussed in this thesis. However, the validation methods introduced later in this chapter will provide an estimate of the accuracy, and this information can in turn be used for maintaining a predetermined accuracy by changing the response time accordingly.

The flexibility of the sensor, i.e. how easy it is to reconfigure it for other purposes, is not addressed directly. Of course, the parametric nature of the suggested algorithm does provide some means for changing the configuration, but flexibility in general is only a

peripheral issue in this thesis. The related parameter versatility is not addressed at all. This also goes for reliability.

The problem of multiple sensors of the same type within range of each other is not intentionally debated in the thesis, but the use of designed signals and orthogonal transforms allows two or more sensors to transmit mathematically independent signals. In general, the solution to the problem is easy when the sensors are synchronized, and a bit more complicated when the sensors are not synchronized.

The remaining parameters reduced size, fault tolerance, and self-calibration are outside the scope of thesis. Intelligence is a somewhat fuzzy parameter, but it is certain that the increased robustness is a (small) contribution to the process of creating an intelligent sensor. And finally, the immunity is obviously increased as well.

4.3.3 Real-Time Signal Processing

The desire for improved robustness by means of signal processing requires the sensor to be capable of doing the signal processing in real-time. This is because the robustness is based on a feedback from receiver to emitter that allows the transmission signal to be adjusted to the current transmission conditions. The feedback consists of information about the most recent transmissions, and this information must be fed to the signal generator relatively quickly if it is to have any value to the signal generating process. The analysis of the received signals and the generation of the transmission signal must therefore be carried out in real-time.

Also, real-time signal processing might easily be relevant in sensor systems where there is no feedback from receiver to emitter. If the receiver needs to respond immediately to changes in the transmission signal (like an abruption of the signal in a through-beam system, i.e. a system where the emitter is facing the receiver from some distance with the purpose of detecting someone or something moving in between the emitter and receiver) real-time signal processing is also necessary.

Requiring that the sensor is able to respond in real-time to unknown occurrences puts a rather strict limit on the amount of processing that can be done to the signal. This limit is given by the signal processing hardware, typically quantified in instructions per second or per sample. In general, the cost of signal processing hardware increases with the computational capabilities, and consequently, the less processing is required the better. The decision on what methods to employ in the algorithm is therefore not only based on desirable mathematical properties, but also on how computational demanding any given method is.

The real-time requirement is on a signal-by-signal basis rather than sample-by-sample. This means that the signal processing is synchronous with the signals, not with each sample in the signals. Each time a signal is recorded it is processed to remove noise, estimate channel gain, determine probabilities etc., and the relevant information is passed to the signal generator which produces an entire signal based on this information.

This means that the timing requirement in hardware as well as software are less strin-

gent than the sampling frequency indicates. For instance, if the sampling frequency in a sensor is 12 kHz and the signal length is 600 samples the signal update rate is 20 Hz. Thus, while the buffers related to the actual transmission and reception are accessed 12.000 times per second by the transmission/reception hardware, they are only accessed 20 times per second by the signal processing hardware. However, in some cases the signal processor also handles the input and output of the digital signals (i.e. holds the two buffers). It is still an advantage to do the signal processing on a signal-by-signal basis, though. It is easier to make efficient use of memory, busses, pointers, etc. when there is a 'deadline' for the signal processing 20 times per second compared to 12.000 times per second.

4.4 Noise and Disturbances

The task of the CGM algorithm is to measure the channel gain between an emitter and a receiver. The reason for using a sophisticated algorithm to accomplish this rather than just emitting a constant signal and measuring the received intensity level is the fact that noise will *always* be present in a real application. The challenge is therefore to design the sensor such that the noise is kept at a reasonable level, and many measures against noise can be taken a priori by carefully designing the casing, circuits, by choosing the right materials, components, etc. The commercial sensor used in the fourth test setup in Chapter 5 is an example of such a design. But it is indeed not possible to completely eliminate the noise.

This section presents the most common types of noise in active sensors. First time and frequency-localized noise is discussed. This is followed by a short introduction to random noise in Section 4.4.2. Finally, in Section 4.4.3 a brief discussion on internal noise in an infrared type of sensors is discussed.

An introduction to some methods for handling the various types of noise is postponed to Section 4.7. To fulfill the requirement for low-cost hardware these methods have to have low computational complexity and thus have to be designed and implemented in close 'collaboration' with the Forward transform and Signal Processing I/II steps in the algorithm, and these are presented in Section 4.5 and 4.6.

4.4.1 Time and Frequency-Localized Noise

A noise occurrence is said to be localized in a given domain if the noise occurs only in a single sample or a few consecutive samples when the signal is represented in that particular domain. The two most common types of localized noise occurs in the time and frequency domain.

Frequency-localized signals are arguably the most common type of localized noise in most environments. Many types of processes generates harmonic signals, purposely or accidentally, and in all sorts of physical forms including electrical, electromagnetic, acoustic, and mechanical. Examples of sources of harmonic signals are loudspeakers, artificial lighting, monitors and displays, electric motors, combustion engines, cellular phones, and remote controls. Many wireless communication systems are also based on

multiplexing in the frequency domain, and thus emit frequency-localized signals into the environment. Any electrical apparatus based on alternating current has the potential of emitting harmonic signals. Note that frequency-localized noise is sometimes referred to as being stationary or non-time-varying. Frequency analysis of signals is a thoroughly researched area, and a huge amount of literature exists in this field.

Any occurrence which is confined to one or a few samples in the received signal is said to be time-localized. Such occurrences are typically not as common as frequency-localized occurrences, although in some environments they appear regularly. Examples of sources of such disturbances are electrical and mechanical apparatuses being activated (or deactivated) causing rapid changes of state. For instance, activating a light source in the presence of an optical receiver causes a sudden change in the signal level and thus a time-localized event. Another example is communication system where the signal is transmitted in bursts to reduce the overall energy consumption. Remote controls are examples of such systems. In fact, the transients generated in the third and fourth test signals in the third test setup, see Fig. 5.18 and 5.19 on page 117 and 118, have been generated with a remote control.

Time-localized noise sometimes exhibits a 'one-way' deviation in the signal, i.e. the affected samples are either above or below the average signal. This happens when short bursts of energy are added to the signal. The circumstances of the conversion from physical to digital signals then determines the sign of the resulting transient.

4.4.2 Random Noise

The noise in a sensor will always be random to some degree as any emitter and receiver component is subjected to quantum mechanical effects, thermal effect, etc. Randomness means that the individual samples cannot be predicted as they are independent of all previous samples. However, often some statistical properties such as amplitude, distribution, and spectral density is known. If the noise samples are not correlated (true randomness) the spectral density is constant and the noise is referred to as being white. This type of noise remains uncorrelated after any orthogonal transformation, but only normally distributed noise maps to the same distribution.

Colored or correlation noise can often be eliminated by taking appropriate (possibly extreme) measures such as electrical and mechanical shielding of the sensor circuits and the channel. This is not possible with the random noise which is caused by physical phenomena such as the randomness of photon emission and the thermal motion of molecules. Although the impact of these effects can be reduced in more controlled environments (for instance by cooling) it is rarely worth the effort as the increase in production costs easily outweighs the performance benefit in most types of sensors.

In the following subsection the light emission uncertainty noise and the thermal noise is quantified for electromagnetic sensors.

4.4.3 Internal Noise in Sensors based on Electromagnetic Radiation

One of the most common physical forms of the signals in active sensors is electromagnetic radiation. Especially visible and infrared light is widely used; the emitter and receiver components are small, robust, fast, and relatively inexpensive. The four test setups presented in the next chapter are all based on infrared light. It is therefore appropriate to include in this chapter a brief discussion of the internal noise conditions for sensor based on electromagnetic radiation. All the information presented in this subsection is from Rogers [65]. Most of the variables and constants used in this subsection are not found anywhere else in this thesis, except for Section 5.4.6 in which the theory presented here is applied to a test setup, and to ease the reading these variables and constants are listed in Table 4.1 rather than being explained in the text.

Table 4.1: Variables and constants used in estimation of noise performance.

Name	Symbol	Unit	Description
Current	i	A	
Power	P	W	
Temperature	T	K	
Radiant intensity	I	W/sr	Radiant flux per unit solid angle of an emitter.
Wave length	λ	m	Wavelength of electromagnetic radiation.
Diode capacitance	C_0	F	Parasitic capacitance of photo diode.
Bandwidth	B	Hz	Bandwidth of photo detection circuit.
Quantum yield	η		Fraction of photons creating electron-hole pairs.
Planck's constant	h	Js	$6.626 \cdot 10^{-34}$
Speed of light	c	m/s	$3 \cdot 10^{8}$
Charge of electron	e	J	$1.602 \cdot 10^{-19}$
Boltzmann's constant	k	J/K	$1.38 \cdot 10^{-23}$

We are seeking to detect a light power of P_r at an optical wavelength λ. Here P_r means the power received by the receiver, not the power emitted by the emitter. This means that $P_\mathrm{r}\lambda/hc$ photons are arriving every second. Suppose that a fraction η of these produce electron-hole pairs (and thus contribute to the generated current). Then there are $\eta P_\mathrm{r}\lambda/hc$ charge carriers of each sign produced every second. The observed electric current is given by

$$i_P = \frac{e\eta P_\mathrm{r}\lambda}{hc} \,. \tag{4.1}$$

Note that the current is proportional to the optical power and to the square root of the electrical power. It is therefore important when specifying the SNR for a detection process, to be sure whether the ratio is stated in terms of electrical or optical power. Apparently, this is a fairly common source of confusion in the specification of detector noise performance [65].

The receiver circuit is divided into the photo detection part and the amplifying part. The amplification is often quite significant and the noise generated in the photo detection part thus becomes the predominant source of internally generated noise. There are basically three types of noise sources in the photo detection part: Shot noise, thermal noise, and dark current. The shot noise is the uncertainty in the arrival of photons from the emitter, the thermal noise is generated in the load resistor, and the dark current is leakage current flowing through the photo diode in the absence of any light input (it is temperature dependent).

The photon emission process in the LED is governed by probability, and thus the photons are emitted randomly. The emitted light intensity is in average a measurable, constant (for constant conditions) quantity, but the random arrival times of the individual particles in the stream imply that there will be statistical deviation from the true value. This deviation must be quantified if the accuracy of the measurements is to be judged. The emission process is Poisson distributed, and it can be shown that this leads to the following shot noise expression for the photo detection

$$i_{\text{shot}} = \sqrt{2eB(i_P + i_{\text{d}})} \ ,$$

where i_{d} is the dark current in the diode and where the bandwidth is given by

$$B = \frac{1}{2\pi R_{\text{load}} C_0} \ . \tag{4.2}$$

In order to gain a true practical appreciation of the noise performance it is necessary to consider the complete photo detection part of the circuit, i.e. to include the thermal noise generated by the load resistor R_{load}. This is given by

$$i_R = \sqrt{\frac{4kTB}{R_{\text{load}}}} \ .$$

Note that i_{shot} expresses the mean of the shot noise current in the sense that the mean noise power is $i_{\text{shot}}^2 R_{\text{load}}$. The same applies to i_R.

The total noise generated by the photo detection circuit is $i_{\text{shot}} + i_R$. The optical SNR in the receiver circuit is therefore given as (in dB)

$$\text{SNR} = 20 \log_{10} \frac{i_P}{i_{\text{shot}} + i_R} \ .$$

In Section 5.4.6 in the next chapter these formulas are applied to a specific test setup.

4.5 Designed Signals and Invertible Transforms

The algorithm presented in Section 4.2 needs an invertible, linear transform to function correctly. The invertibility requirement is necessary to use the transform in the fashion

presented earlier, i.e. where the signal is transformed from one domain to another to suit the transmission conditions, and then transformed back to the first domain. The linearity is required to facilitate an easy post-processing (this will be evident in Section 4.6 on estimation of channel gain).

A prerequisite for applying the transformation is having a signal to apply it to. And if the outcome is to be a signal with certain properties needed for a robust transmission the original signal must be carefully designed. Of course, the knowledge of the behaviour of the transform is important information when designing the signal, but the idea of the algorithm is to choose a transform which is well-suited for the given scenario such that the designed signal is simple and straightforward to design, and such that a few parameters in this design control all the important factors of the signal structure. The concept of designed signals is introduced in Section 4.5.1. The design signal concept goes beyond the transmission signals. A set of test signals is also designed. These serve a number of purposes which are discussed in Section 4.5.2 and 4.7.4.

The choice of transform depends very much on the sensor application and there exists a variety of transforms with a diversity of properties. They all fall into two basic categories, though. The time-frequency localizing transforms and the time-frequency spreading transforms. In Section 4.5.3 some of the most important aspects of choosing transform is discussed. In many cases it is recommendable to use an orthogonal transform due to its nice properties, but occasionally the orthogonality must be skipped in favor of other more important properties. In Section 4.5.4 it is discussed whether to choose an orthogonal transform or not.

4.5.1 Concept of Designed Signals

One of the challenges when constructing an active sensor is making a list of transmission signals that will be suitable in the various conditions which the sensor will be operating in. The signals should be such that they are not easily confused with typical noise occurrences and such that post-processing is easy to do. They have to comply with the limitations of the sensor hardware such as finite precision and transfer function of amplifiers. In multi-sensor systems the signals should also be easy to separate and in systems where the emitter and receiver are not connected synchronization should be possible by using just the signals.

It is not impossible to get signals which satisfies all of the above mentioned properties, at least to some extent. However, it is clear that it would be cumbersome, if not virtually impossible, to make a list of signals by hand and then store each of them in the sensor. The solution is to have a systematic method for creating signals on the fly and with parameterized properties. The former enables the sensor to respond quickly to changed conditions while the latter makes it easy to achieve a variety of properties in the signals by simply adjusting the parameters.

While there undoubtedly exist a number of ways to systematize the construction only one approach is discussed in this thesis. The basic idea is to use invertible transforms

in combination with designed signals. Two versions of this idea was presented in Section 4.1.1 and 4.1.2. The overall structure and properties of the constructed signals are chosen via the choice of transform, while the parameterization is handled by the design of the signals prior to transformation. Occasionally, the transform includes one or more parameters also. The designed signals are kept relatively simple since this makes the post-processing easier and less computational demanding. The use of transforms to generate the signals is to a large extent an automation of the whole sensing process. For by using a transform it also becomes easier to gather useful information about the sensor environment and to make decisions about the behavior of the sensor. This is explained in more detail in Section 4.6.

The chosen transform is responsible for providing the properties listed in the beginning of this subsection. These properties form the basis for the high degree of ease and automation just mentioned. The aspects of the inverse transform is discussed in more details in the subsections Section 4.5.3 and 4.5.4.

The total computational load of the CGM algorithm is somewhat higher than an algorithm which uses a set of predetermined signals, but the benefits outweighs this disadvantage for two reasons. Firstly, the algorithm has the potential of becoming much more adaptable, robust, and fault tolerant. Secondly, the computational load of the CGM algorithm is still small compared to the computational power offered by existing signal processing hardware.

4.5.2 Concept of Test Signals

Robustness is one of the important aspects of sensor design in the context of this thesis. A high robustness is achieved partly by constructing transmission signals which are well suited for the sensor environment and partly by estimating to what extent the signal has been distorted or corrupted during transmission. Since the transmitted signals are known there is indeed potential for obtaining a good estimate.

One way to do this was presented in the descriptions of the two embodiments earlier in this chapter. The formulation was that a number of channels are available, and only some of them are used for transmission while the remaining are used for detecting noise. Here the same idea is formulated in terms of linear algebra.

Let N be the number of samples in the designed signals and in the transmission signal. For the sake of convenience it is assumed that only one transmission signals is needed (this has mainly a bearing on the notation). The signals which after transformation becomes the transmission signals is called the designed signal or original, designed signal, depending on the context. This signal is $\mathbf{u}_0$. It is possible to construct $N-1$ other signals which are linearly independent of each other and of $\mathbf{u}_0$. These are denoted $\mathbf{u}_1$ through $\mathbf{u}_{N-1}$. Though it is not strictly necessary to require these signals to be orthogonal it does make some calculations as well as interpretations of results easier. It is therefore assumed in the following that the signals are indeed orthogonal.

The purpose of designing many signals instead of just one is to have an easy and

automated way of assessing the transmission noise. When orthogonal the signals have the property that the inner product between the received, transformed signal, denoted $\mathbf{y}$, and $\mathbf{u}_n$ is $\delta[n]$ for an ideal transmission, and that the amplitude of $\langle \mathbf{y}, \mathbf{u}_n \rangle$, $n \geq 1$, indicates to what extent the transmission has been distorted. Consequently, an estimate on the form

$$p = \frac{\langle \mathbf{y}, \mathbf{u}_0 \rangle}{\sum_{n=0}^{K} | \langle \mathbf{y}, \mathbf{u}_n \rangle |} , \tag{4.3}$$

provides number between 0 and 1 which is a good indication of the noise level in the transmission. Variations of this form is indeed possible, and for instance the validation method described in Section 4.9.6 employs a somewhat different form of this estimate.

4.5.3 Choosing the Transform

It is important to choose the right transform for the algorithm. This is evident from the introduction to the concept of designed and test signals in the previous two subsections. It also became clear that the transform should possess a number of properties to ensure

- easy design of transmission signals,
- easy post-processing,
- easy estimation of different types of noise contributions,
- possibility for separation of signals in multi-sensor systems,
- high numerical stability,
- low computational complexity, and, of course,
- that the transmission signal can be constructed to suit the transmission conditions.

As the last property is essential for a successful channel gain measurement it has a top priority. A good transmission signal is either easily distinguishable from or has a low sensitivity to the most common noise contributions. Since it is assumed that for the sensor systems described in this thesis the three most common types of noise is white noise, and time and frequency-localized noise, the chosen transform should be able to isolate (localize) these noise types or reduced the sensitivity to these noise types. When disregarding the white noise this can be achieved by joint time-frequency (JTF) transforms while the latter can be achieved by spread spectrum (SS) transforms. A white noise contribution cannot be localized by any linear transform, though it can be reduced in predetermined parts of the domain (which is what happens with band pass filtering).

The JTF transforms includes all transforms which yields some kind of separation of the signal energy in time and frequency. Examples of such transforms are wavelet and wavelet packet transforms, Gabor transform (Pedersen [63], Zielinski [84]), short-time Fourier transform (Qian and Chen [64]), local trigonometric transform (Auscher et al. [3], Hernández and Weiss [41]) and time-varying modulated lapped transforms (Vetterli and Kovačević [79], Malvar [56]), Wigner transform (Claasen and Mecklenbrauker [18, 19, 20]), and, in general, transforms from Cohen's class, see again Qian and Chen [64].

While a JTF transform is capable of analyzing the distribution of energy in time and frequency it is not, as a consequence of Heisenberg's uncertainty relation, see for instance Gröchenig [35], capable of perform the analysis with arbitrary accuracy. Consequently, it is necessary to accept a trade-off between resolution in time and frequency. For some transforms this trade-off is easy to adjust, and in many cases tricks exist for obtaining particular properties in the time or frequency domain. An example is the WPT where increased smoothness of the wavelet filter tends to degrade the time localizing ability. By using a set of different, carefully designed wavelet filters throughout the decomposition it is possible to retain the time localizing ability while preserving a certain degree of smoothness, see Selesnick [69].

The spread spectrum transforms distributes the energy in the signals approximately evenly throughout the spectrum, i.e. in the frequency domain. The spread spectrum technique has been widely used in the last few decades. Most notably, the global positioning system and CDMA in mobile telephony, see Viterbi [81], employs spread spectrum signals. Also, a number of other communication systems employ spread spectrum modulation, see for instance Kesteloot and Hutchinson [48] and Simon et al. [71]. There exists a variety of methods for creating spread spectrum signals, each having properties suited for particular scenarios. In this thesis the focus is solely on the Rudin-Shapiro transform, in some literature denoted PONS (Prometheus orthonormal set), see Byrnes et al. [14]. There is a large number of references to Rudin-Shapiro related literature in Chapter 11, but none of these references introduce the spread spectrum technique from an engineering point-of-view. However, the nice tutorial on spread spectrum signals by Viterbi [82] does. And more thorough presentations of spread spectrum systems is Dixon [30] and Cooper and McGillem [25].

The easy design of transmission signals is a property which is totally dependent on the interpretation of the transform. Obviously, any signal can be generated by any linear, full rank transform with the right original, designed signal, but the idea is to have a transform which allows suitable transmission signals to be generated by simple, designed signal. By choosing a JTF or SS transform it becomes easy to design JTF and SS sequences, respectively, and this provides the potential for parameterizing the construction of transmission signals.

4.5.4 Orthogonal and Biorthogonal Transforms

Of the many properties listed in the beginning of the previous subsection only two are addressed, namely the ability of the transform to constructed a signal suitable for the transmission conditions and easy design of transmission signals. An easy way to get a big step closer to having a transform with these properties is restricting the choice to an orthogonal transform (alternatively, the transform might be required to be unitary, as this is equivalent to orthogonal for real matrices). Although it is indeed possible to choose an orthogonal transform which would not be useful at all (a Gram-Schmidt orthogonalization of a random matrix is an example) it is nevertheless a very helpful restriction from an

applicational point-of-view. This is because some of the properties are guaranteed with an orthogonal transform. This includes easy estimation of white noise level, separation of signals, and numerical stability. The separation of signals is obviously easy when cross-terms between transmitted signals are zero, and the overall numerical stability is provided by the fact that an orthogonal transform is energy preserving. Of course, there might be numerical problems in a particular implementation if the intermediate calculations involve very large and very small numbers simultaneously. An introduction to the concept of orthogonal transforms in signal processing is found in Ahmed and Rao [1].

The easy estimation of white noise level is basically due to the fact that a white noise contribution stays white under orthogonal transformation. Although it seems obvious that any reduction of white noise caused by transformation is desirable, this is not the case. This is because any transform is perfectly localizing in its 'own' domain, e.g. the RST is perfectly localizing in the domain of RS sequences, and since the transmitted signals is known, only a single sample holds the information on the channel gain (this is also true for the orthogonal JTF transforms, see the next paragraph) while the remaining samples are only noise. Since the noise stays white the effect on a single sample is limited to $1/N$'th of the white noise energy. Obviously, it does not make sense to reduce the noise on the samples which do not hold any energy from the transmitted signal. Instead, if the noise is preserved, these samples can be used to estimate the statistical properties of the noise, and thereby provide an estimate of the accuracy of the one sample that represents the CGM.

Note that the set of test signals introduced in Section 4.5.2 also constitutes an orthogonal transform. Here the orthogonality means that it is easy to relocate the entire energy of the transmitted signals into a single sample. This happens automatically for RS sequences, but not, in general, for JTF transforms. The concentration of the energy means that the 'true' CGM can be measure in a single sample and that the remaining samples are just noise. If the test signal transform was not orthogonal the noise samples would be correlated with the signal making the post-processing more complicated.

In some cases it might be beneficial to abandon the orthogonality requirement despite the nice properties that comes with an orthogonal transform. For instance, it is not possible to have a finite, symmetric, orthogonal wavelet filter, and consequently, any orthogonal wavelet transform will not map symmetric signals to symmetric signals. Also, it is in general easier to handle the edge problem (see Chapter 9 and 10) with symmetric filters. An alternative is biorthogonal transforms where one basis is used to forward transformation and another basis is used for inverse transformation. The two basis sets are orthogonal to each other, but they are not orthogonal sets in themselves. Biorthogonal transforms are well-known in the field of wavelets (they sometimes appear under the term 'frames'), where many applications use this relaxed version of the wavelet transform. Most books on wavelets include a chapter on biorthogonal wavelets. See for instance Chui [17, Ch. 5] and Burrus et al. [10, Sect. 7.4]. Note that since Parseval's theorem no longer holds in the biorthogonal case there is a potential danger of an exponential growth in sample amplitude. As severe numerical instability could be the result this concern

should be addressed when using a biorthogonal transform.

4.5.5 Applicational Properties of the Transform

The previous subsections have discussed a number of important mathematical properties of the transform. However, a transform might have a series of nice and desirable mathematical properties and still be useless in a low-cost sensor system. In order to apply a transform to a real world problem it has to have a number of applicational properties, too. It is required that the transform is

- numerically stable,
- subject to a sensible interpretation,
- easy to implement,
- flexible,
- computationally not too complex, and
- applicable to finite signals.

These requirements are related to the combination of transform and signal processing hardware rather than to the mathematical properties, and ensure that the transform can indeed be implemented and used in a low-cost sensor. The requirements are discussed one by one in the following.

The numerical stability is an obvious requirement in fixed point signal processing hardware. The relative low precision makes the transform vulnerable to rounding errors and overflow if the transform has a large dynamical range on the intermediate calculations. The presence of noise also adds to the problem if the transform is numerically unstable.

The basic vectors in the transform matrix must have an interpretation which is in line with the expected type of noise. For instance, if frequency-localized noise is expected to be dominating, and some isolated, time-localized noise burst are expected, too, the transform should have an balanced interpretation in the frequency and time domains. That is, the basic vectors must be frequency localizing to some extent without completely sacrificing the time resolution. A good interpretation allows the control part of the sensor to easily detect the noise as well as easily change the design signal to avoid the noise. This can be described as finding and exploiting ‘holes’ in the noise. This becomes quite difficult if the noise does not have a simple interpretation in the given basis.

The only requirement which can be ignore without major consequences is the easy implementation. This is also sometimes referred to as low programmable complexity. While it is an advantages to have a simple transform structure when programming it is not essential for the implementation. Note that the RST as well as the WPT have quite simple structures.

Flexibility means that the transform can easily be changed to accommodate shorter or longer signals, can be implemented in many types of hardware, is parameterized in a way which enables easy adjustment to new applications, and so on.

While all linear transforms can be implemented with complexity $O(N^2)$ it is rarely acceptable. Fortunately, in many cases an $O(N \log N)$ implementation exists. The difference between such two implementations is significant for longer signals, and the existence of a good implementation can be the decisive difference between two transforms. The most famous example of a fast implementation is the fast Fourier transform (FFT) presented by Cooley and Tukey [24]. The WPT and the RST are also fast transforms; both has complexity $O(N \log N)$. Examples of transforms with no fast implementation (to the best of the authors knowledge) is the Gabor transform and Wigner transform.

Finally, a requirement which seems obvious, but which nevertheless tends to be ignored until the actual implementation takes place. The transform should be able to handle finite signals in an appropriate manner. Especially short signals cause problems for some transforms. In the WPT it is a challenge to find a good method for handling the edges of the signal (see Chapter 9 and 10) while the RST does not have any such problems as it is based on two tap filters.

4.6 Estimating the Channel Gain

The very purpose of the algorithm is to estimate the channel gain in order to provide the sensor functionality and this step in the algorithm is therefore of great importance. Yet, it is the least computational demanding step, as will be evident by the end of this section. First, the formulation of the transmission signals is given in given in Section 4.6.1 and the result is subjected to a least square analysis in Section 4.6.2 to provide an estimate of channel gain as well as noise.

4.6.1 Transmission of the Signal

The starting point is the original, designed signal $\mathbf{u}_0$ which is transformed by $\mathbf{W}^{-1}$ and adjusted by an affine mapping to fit the DAC (to ease notation it is assumed that $\mathbf{W}$ is square). The resulting signal is

$$\mathbf{t} = \alpha \mathbf{W}^{-1}\mathbf{u}_0 + \beta \mathbf{1}.$$

The transmitted signal is given by $\mathbf{x} = T(\mathbf{t})$, where T is the transfer operator from emitter to receiver, including component characteristics, cross talk, non-linearity in amplifier etc. It is assumed that T is either a constant transfer function or a known transfer function (except for the gain). It is also assumed that the transmission dampens the signal and adds noise, that is $T(\mathbf{x}) = G\mathbf{x} + \mathbf{e}_\mathrm{t}$. Note that a subscript 't' has been added to $\mathbf{e}$ to avoid confusion with the canonical basis vectors $\mathbf{e}_n$. The forward transform $\mathbf{W}$ of the received signal yields

$$\begin{aligned} \mathbf{y} &= \mathbf{W}\mathbf{x} \\ &= \mathbf{W}(G(\alpha \mathbf{W}^{-1}\mathbf{u}_0 + \beta \mathbf{1}) + \mathbf{e}_\mathrm{t}) \end{aligned}$$

$$= G(\alpha \mathbf{u}_0 + \beta \mathbf{W1}) + \mathbf{W}\mathbf{e}_t, \tag{4.4}$$

where G is the channel gain. The terms $G\beta\mathbf{W1}$ and $\mathbf{W}\mathbf{e}_t$ are both unwanted signal components. The last term is unwanted simply because it is pure noise. The middle term is unwanted because although it does theoretically assist in determining G it will usually resemble the low frequency noise (this is the case with the RST method as well as the WPT method) and it will usually have a significant amount of energy compared to the first term. Depending on the transform this component can be removed one way or the other. For instance, when $\mathbf{W}$ is the RST $\mathbf{W1}$ is a constant signal which can be completely removed simply by subtracting the mean of the whole signal (as this is a linear operation).

The gain G represents the total damping of the signal from it left the signal processor and until it is back in the signal processor. In the vast majority of sensor applications all the quantities in the entire physical setup of the sensor is not known, and G then becomes a relative gain rather than an absolute gain. This means that a particular value of G does not have a meaningful interpretation, but a variation in G does. Consequently, the α in (4.4) can be considered a part of G without loss of information.

As a consequence, the received, transformed signal $\mathbf{y}$ can for the purpose of estimating the channel gain be regarded as being on the form

$$\mathbf{y} = G\mathbf{u}_0 + \mathbf{e}_{wt}, \tag{4.5}$$

where $\mathbf{e}_{wt}$ is the transform of the noise component $\mathbf{e}_t$. Available in this equation are three important degrees of freedom; the choice of original signal, the choice of transform, and the choice of solution method. While the two first are intimately related the choice of solution method is to a large extent independent of the first two degrees of freedom.

4.6.2 Estimating Gain and White Noise with Linear Equations

The vector equation (4.5) can be considered as a system of N linear equations with $N+1$ unknowns; the entries of the noise vector, and the gain. The size N of the system depends only on the number of non-vanishing coefficients in the original signal $\mathbf{u}_0$ and the chosen transform. Since the coefficients on G is directly controllable via $\mathbf{u}_0$ the linear equation system can be tailored to fit an approximate solution method such as least squares. If the noise is normally distributed, $\mathbf{e}_t \sim N(\mu, \sigma^2)$, this will give the best result, independently of whether it is applied before or after a orthogonal transformation since $\mathbf{e}_{wt} \sim N(\mu, \sigma^2)$ if and only if the same applies to $\mathbf{e}_t$. A least squares approach could be formulated through a rewriting of (4.5) to

$$\|\mathbf{y} - G\mathbf{u}_0 - \mu\mathbf{1}\|^2 = \sigma^2 N, \tag{4.6}$$

where N is the length of the signal. This is rewritten to

$$\|\mathbf{y}\|^2 + G^2\|\mathbf{u}_0\|^2 + \mu^2 N^2 - 2G\langle \mathbf{y}, \mathbf{u}_0\rangle - 2\mu\langle \mathbf{y}, \mathbf{1}\rangle + 2G\mu\langle \mathbf{u_0}, \mathbf{1}\rangle = \sigma^2 N, \tag{4.7}$$

The left hand side of (4.7) is an elliptic paraboloid in μ and G that opens upwards and with minimum for some value σ^2. This is the best estimate of the variance, and for exactly this

value only one set of (G, μ) satisfies the equation. This minimum point is found when the G and μ derivatives are zero simultaneously. Solving that yields

$$G = \frac{\langle \mathbf{y}, \mathbf{u_0} \rangle N - \langle \mathbf{u}_0, \mathbf{1} \rangle \langle \mathbf{y}, \mathbf{1} \rangle}{\|\mathbf{u}_0\|^2 N - \langle \mathbf{u}_0, \mathbf{1} \rangle^2}, \tag{4.8}$$

$$\mu = \frac{\langle \mathbf{y}, \mathbf{1} \rangle \|\mathbf{u}_0\|^2 - \langle \mathbf{u}_0, \mathbf{1} \rangle \langle \mathbf{u}_0, \mathbf{y} \rangle}{\|\mathbf{u}_0\|^2 N - \langle \mathbf{u}_0, \mathbf{1} \rangle^2}. \tag{4.9}$$

The smallest σ is then

$$\sigma^2 = \frac{\|\mathbf{u}_0\|^2 \|\mathbf{y}\|^2 N - \|\mathbf{u}_0\|^2 \langle \mathbf{y}, \mathbf{1} \rangle^2 - \langle \mathbf{y}, \mathbf{u}_0 \rangle^2 N - \|\mathbf{y}\|^2 \langle \mathbf{u}_0, \mathbf{1} \rangle^2 + 2 \langle \mathbf{y}, \mathbf{u}_0 \rangle \langle \mathbf{u}_0, \mathbf{1} \rangle \langle \mathbf{y}, \mathbf{1} \rangle}{\|\mathbf{u}_0\|^2 N^2 - \langle \mathbf{u}_0, \mathbf{1} \rangle^2 N} \tag{4.10}$$

When these equations are subjected to the assumption that the zeroth moment of $\mathbf{u}_0$ is vanishing they reduce to

$$G = \frac{\langle \mathbf{y}, \mathbf{u}_0 \rangle}{\|\mathbf{u}_0\|^2}, \tag{4.11}$$

$$\mu = \frac{\langle \mathbf{y}, \mathbf{1} \rangle}{N} = \frac{1}{N} \sum_n y_n, \tag{4.12}$$

$$\sigma^2 = \frac{\|\mathbf{y}\|^2 - G \langle \mathbf{y}, \mathbf{u}_0 \rangle}{N} - \mu^2, \tag{4.13}$$

which are the expected formulas when the influence of the $\mathbf{u}_0$ signal is 'neutralized'. Recall that if $E(y)$ is the expected value of y then $\sigma^2 = E(y^2) - E(y)^2$. As long as the noise is normally distributed no means for providing a better estimate of G exists. But often the noise is not normally distributed, or at least consists of a white noise contribution as well as a colored noise contribution. In that case the given estimates of G, μ, and σ^2 can be very misleading. In the next chapter these estimates are made for a number of different signals, in particular Table 5.5 on page 5.5 presents the accuracy of the estimates on three signals. However, since in all experiments throughout the next chapter the mean of $\mathbf{u}_0$ is zero these estimates are merely the well-known formulas for mean and variance.

The misleading estimates of G, μ, σ^2 in colored noise makes it interesting to have other methods for determining the properties of the noise. In particular, it would be nice to be able to remove all colored noise from the signal as this will make (4.11) the best estimate of G. As described in Section 4.4 two very common types of disturbances is time and frequency-localized noise. The following section therefore focuses on methods for removing these types of noise.

Sometimes it is not possible to perform a sufficient denoising within the limitations of the hardware and the algorithm. This may be because no resources have been allocated to denoising, or because the noise occurrence does not fit the chosen denoising methods.

In such cases the best one can do is estimating how bad, i.e. to what extent, the noise occurrence has affected the CGM. By having a threshold (or some other means) this estimate can be turned into a validation of the CGM. Two methods for detecting and validating CGMs are discussed in Section 4.9.

4.7 Denoising

There are various forms of denoising which can be applied to the received signal. This and the following section present a number of suitable methods. This section focuses on the traditional means, while the next sections discusses in details how to use polynomials for removing low frequency noise in relation to the two previous embodiments (especially the spread spectrum method).

4.7.1 Frequency-Localized Noise

The presence of frequency-localized noise is a problem is virtually all applications employing signal processing. There exists many method for removing, or at least reducing, this noise, and this section will address only a few of these, namely

- Single frequency approximation
- Band pass filtering
- Wavelet decomposition
- Polynomial approximation

The first two are traditional, electrical engineering methods for noise denoising, and are only discussed briefly. The wavelet decomposition is in the frequency interpretation a set of band pass filters, and the effectiveness of these are discussed. Finally, a decomposition into a polynomial basis is discussed in Section 4.8 as a method for removing low frequency noise in the spread spectrum case.

A common case of frequency-localized noise is a single, dominant frequency, e.g. sinusoid in the signal. If the frequency is known the brute force way of removing such a frequency is the following. Let s be a finite, continuous signal on the form

$$s(t) = \sum_{n\in\mathbb{N}} a_n \sin(nt + \phi_n), \quad t, \phi_k \in [0; 2\pi) \ . \tag{4.14}$$

Then $s \in L^2\big([0; 2\pi)\big)$ whenever $\mathbf{a} \in \ell^2(\mathbb{N})$. The signal component represented by $a_k \sin(kt + \phi_k)$ can be separated from the signal in the following way. Since

$$\int_0^{2\pi} \sin(mt + \phi_m) \sin(nt + \phi_n) dt = \delta[m - n]\pi \cos(\phi_m - \phi_n), \quad m, n \in \mathbb{N}$$

it follows that

$$\langle s, \sin(k\cdot)\rangle = a_k\pi \cos(\phi_k) \qquad \text{and} \qquad \langle s, \cos(k\cdot)\rangle = a_k\pi \sin(\phi_k) \ ,$$

with the inner products defined for $L^2\big([0;\,2\pi)\big)$. For a finite, sampled signal these inner products can be estimated with an accuracy determined by the ratio of sampling rate and the desired frequency. By subtracting

$$\begin{aligned} a_k \sin(kt+\phi_k) &= a_k\big(\sin(kt)\cos(\phi_k)+\cos(kt)\sin(\phi_k)\big) \\ &= \frac{\langle s,\sin(k\cdot)\rangle}{\pi}\sin(kt)+\frac{\langle s,\cos(k\cdot)\rangle}{\pi}\cos(kt) \end{aligned}$$

from the signal $s(t)$ this particular frequency has been removed. Although this approach is theoretically sound it may not be the best way of approximating a single frequency. A number of ways have been investigated, see for instance Kay [47] and Klein [49].

An alternative to singling out a particular frequency (which has to be a priori known) is band-stop filtering which targets a range of frequencies. Low frequency noise can effectively be removed by a high pass filter which can be designed to fit any predetermined break frequency and with any Q factor. Targeting a particular frequency (or at least a very small range of frequencies) is possible with a notch filter. This is given as $(s^2+\omega_0^2)/(s^2+\xi s+\omega_0^2)$ and a very small ξ yields a transfer function which is almost constant except in a neighborhood of ω_0. In some cases this approach is very useful, but bearing in mind that the suggested algorithm is based on analysis of blocks of samples (referred to as signals) a good filter becomes less attractive to use as a part of the denoising process. In particular, a notch filter is an IIR filter with slowly decaying taps, and thus it is only effective when many consecutive samples are available.

When a JTF localizing transform is used in the algorithm the band pass filtering is an inherent part of the process and usually no other filtering is necessary in this case. This is discussed in the next subsection. When a spread spectrum transform is used there is no built-in parameterized reduction of particular frequencies (however, a SS transform does necessarily suppress some fixed frequencies, see Section 11), and consequently any need for removing particular ranges of frequencies must be handled outside the transform. In that case a band pass filtering is one solution. Since band pass filtering is a well-understood method and a huge amount of literature exists on the subject it is not discussed in further details in this thesis.

As stated in the beginning of this section another solution is removal of low degree polynomial content. On sufficient short intervals low frequencies can be well approximated by a low degree polynomial and thus removed. This approach is introduced in details in Section 4.8.

4.7.2 Frequency-Localized Noise in the Wavelet Packet Decomposition

When the wavelet packet decomposition is interpreted in the frequency domain it is revealed as a set of band pass filters. This is basically because the two wavelet filters used in the decomposition is a low and a high pass filter. The details of this interpretation can be found in most text books on wavelets, see for instance Jensen and la Cour-Harbo [45],

Vetterli and Kovačević [80], and Daubechies [26]. There are a number of issues that needs to be addressed if one wants to explicitly use the frequency related properties of the WP transform. These will not be discussed here, as the mentioned literature covers these issues. However, it is interesting in the context of this thesis to learn how good the frequency localizing property of the transform is. This is the case not only in respect of removing low frequency noise, but also when the designed signal is used through the inverse wavelet transformation to generate a transmission signal with certain frequency properties.

The quality of the various wavelet filters are discussed in the mentioned literature, and one can choose whatever filter is believed to be most suitable for a given application as the WPT structure is independent of the filters (the edge handling procedure is not, however). No fairly short FIR wavelet filter has frequency localizing properties which comes close to what can be achieved with other types of filters. An attempt to optimize frequency localization was done by Hess-Nielsen, see [42] and [43].

Fortunately, the comparatively low quality factor of the filters is rarely a problem. This is because the search for 'holes in the noise', as explained in Section 4.1.2 and 4.5.3, is not directly aimed at finding low-noise frequency ranges, but rather at finding low-noise ranges in the transform domain. When the transform domain then 'happens' to be a frequency localizing domain, the low-noise parts will also be low-noise frequency ranges. It is important to realize that the best basis algorithm, which would typically be used for finding the holes in the noise, does not rely on the frequency interpretation, and therefore gives the best estimate of distribution of noise in the transform domain regardless of the fact that WPT might delivered a frequency separation of the signal which is far from being the best achievable.

To get an impression of the frequency response of a typical wavelet filter Fig. 4.4 shows the frequency content of a designed, inversely wavelet transformed signal, where the energy has been put into the fifth of eight elements on the fourth level of a WP decomposition. Note that since every element represents the whole time line of the signal, setting the signal to a constant non-zero value in one element prior to transformation will yield a signal with a rather narrow frequency content. To use the full potential of the frequency band corresponding to that particular element a spread spectrum sequence is used instead. In Fig. 4.4 a sampled chirp has been used, but any SS sequence will do, including an RS sequence.

In the same figure the frequency response of a three times iterated Symlets 6 tap filter is shown, too. This filter has been used in the wavelet transformation of the designed signal. Note how the frequency response is quite broad compared to the ideal filter. In particular, the response is asymmetric with a side loop to the right.

Another example is shown in Fig. 4.5. Here the frequency response of the three times iterated Daubechies 12 tap and CDF(4,6) filters are shown in all eight bands. Note that the vertical scale is now linear. While the frequency response of the orthogonal Daubechies 12 is acceptable, in particular the lowermost band pass, the response of the biorthogonal CDF(4,6) is rather poor. One should be careful using this filter for frequency related

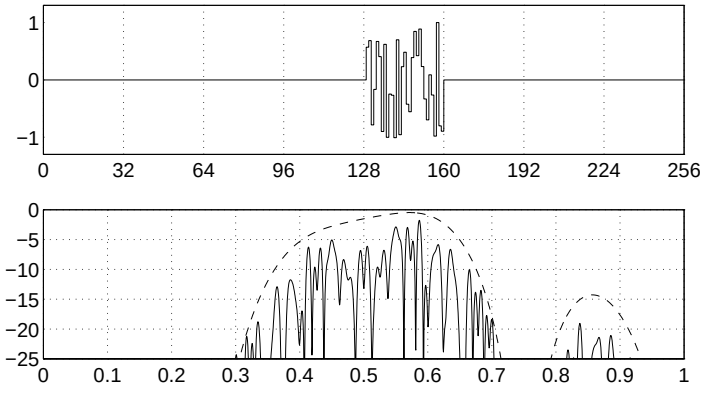


Figure 4.4: The top plot shows a designed signal which under a three level inverse WP transformation will give a signal with energy approximately in the frequency band $4\,f_s/16$ to $5\,f_s/16$. The samples in the interval [128; 160] are generated by sampling a chirp. The lowermost plot shows in solid line the actual frequency content of the signal after inverse transformation and in dashed line the 5th sub-band of a 3 times iterated Symlets 6 filter. The second axis is relative dB.

denoising.

One important observation is that low frequency noise will indeed be handled appropriately by the wavelet transform in the sense that low frequency content will appear in only one element after transformation. Consequently, it is not necessary to apply any extra filtering in case of low frequency noise when using the wavelet modulation method for generating and post-processing signals.

4.7.3 Time-Localized Noise in JTF and SS Transforms

The presence of time-localized noise (transients) in the received signal yields two very different results when subjected to a JTF transform and an SS transform, respectively. While the former keeps the transient energy in relatively few samples the latter by construction spreads the energy more or less evenly on all samples in the transformed signal. Since this spreading effect as an alternative to denoising is one of the reasons for choosing an SS transform, an attempt to denoise an SS signal obviously somewhat obscures the point of choosing such a transform. This is not to say that denoising would not increase the accuracy of the CGM, but the ratio between denoising effort and increased accuracy is higher for SS signals than for JTF signals. Consequently, the time-localized denoising is only considered for JTF transforms.

The claim that it is sensible to put an effort into removing transients from a JTF modulated signal, but not from a SS modulated signal is perhaps a bit more subtle than one

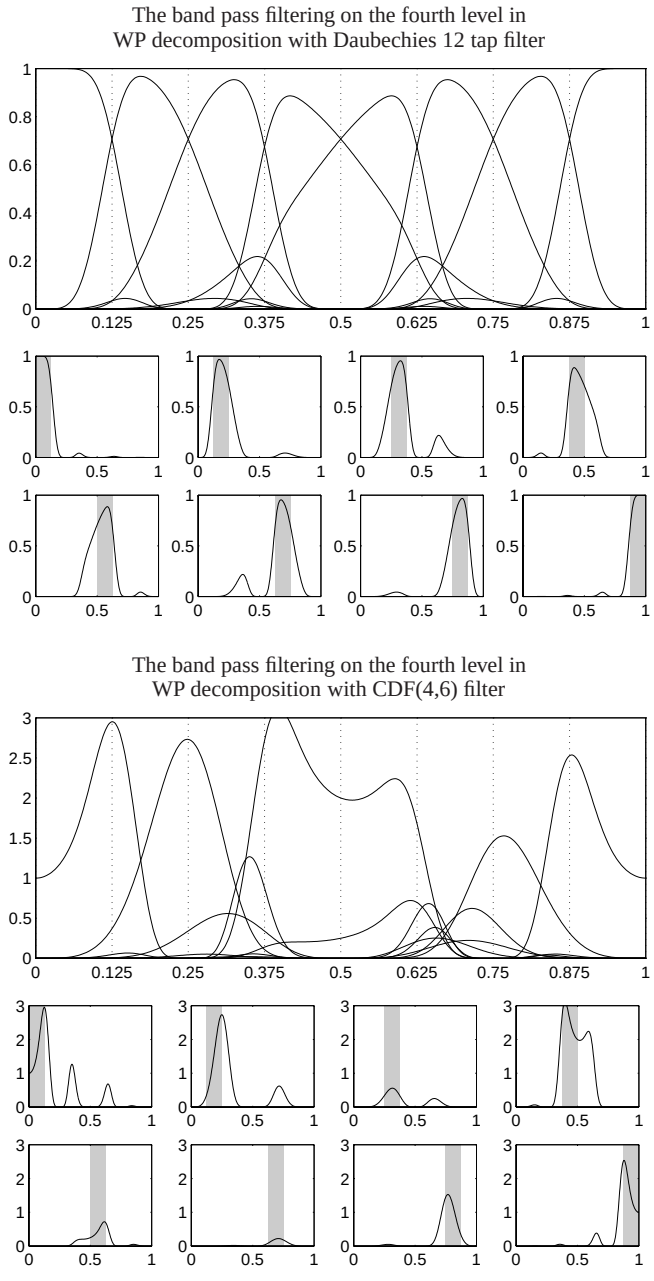


Figure 4.5: The band pass filters for three times iterated Daubechies 12 and CDF(4,6) filters. The frequency response of each filter is shown first in a shared plot and below individually with the corresponding frequency band in gray. Source: Jensen and la Cour-Harbo [45].

would immediately realized from the above description. This is because of the following calculation, which argues that the impact of a transient on the CGM is the same for the RST and WPT, i.e. that the estimate of CGM has a certain accuracy independently of whether the wavelet or RS modulation has been used.

Assume that a modulated signal has been transmitted and only one of the 2^J samples in the signal has been subjected to a disturbance (white noise is disregarded as this arguably has equal impact under any orthogonal transform). Assume also that the disturbance is additive noise with energy E^2. If the transmitted signal is RS modulated the RST will produce a signal where each sample, including the one providing the CGM, has been affected (additively) by $E/2^{J/2}$, since

$$\sum_{n=0}^{2^J-1}\Big(\frac{E}{2^{J/2}}\Big)^2 = E^2 \, .$$

If the transmitted signal is wavelet modulated the WPT will produce a signal, where each element in the chosen basis will have one significant non-vanishing entry (except for the element representing the channel chosen for transmission where all entries are non-vanishing). Since the orthogonal WPT is energy preserving (the biorthogonal is often almost energy preserving) the total energy of the transients is E^2. A transient is presented at all frequencies and therefore the transients, one in each element, have an amount of energy approximately reversely proportional to the number of entries in the element. Consequently, a transient in a element on level j (starting index 0) has an energy approximately equal to $E/2^{j/2}$, since

$$\sum_{j\in A}\Big(\frac{E}{2^{j/2}}\Big)^2 = E^2\sum_{j\in A}2^{-j} = E^2 \, ,$$

where A is a list of the level number of each element (0 being the top level, i.e. the un-transformed signal). The following inner product, as specified in (4.11), applies approximately the same weight to all samples. For a unit energy designed signal $\mathbf{u}_0$ this weight is $2^{(j-J)/2}$, and thus the transient affects the CGM by

$$2^{(j-J)/2}\frac{E}{2^{j/2}} = \frac{E}{2^{J/2}} \, .$$

This argument goes to show that there is in general nothing gained in terms of CGM accuracy under time-localized disturbances by employing the WPT rather than the RST (or vice versa). This obviously raises the question why one transform is preferable to the other in a case where the time-localized noise is expected to be dominant. The answer is twofold: The WPT is not representative for all JTF transforms (see the two next paragraphs), and detecting and removing transients is easier in a JTF transform scenario than in a SS transform scenario (see the next section).

There exists many different JTF transforms that behave in different ways and produce outputs in different formats, see Qian and Chen [64]. Not all JTF transforms have a filter

bank output format like the WPT. For instance, the short-time Fourier transform (STFT) and the local trigonometric transform (LTT) both have a segmentation of the output in time domain, and each segment represents the entire frequency range, whereas the WPT has a segmentation in the frequency domain where each segment represents the entire time line (of the transformed signal).

When a signal with a transient is transformed with a frequency segmenting JTF, like the WPT, the transient reappears at approximately the same time location in each segment. When the same signal is transformed with a time segmenting JTF the transient affects all the sample in the segment which corresponds to the location in time of the transient. This means that while there is arguably no difference between the influence of a transient on the CGM in the RST and the WPT cases, there is indeed if the JTF transform is the LTT. Either the CGM estimate is distorted by the entire energy of the transient (when time segment chosen by means of the original designed signal overlaps the transient) or there is no influence at all (when the time segment does not overlap the transient).

4.7.4 Detecting and Removing Time-Localized Noise in a JTF Transform

The presence of high energy time-localized noise, transients, in the received signal is not difficult to detect. Such noise is by definition concentrated on a relatively small number of samples which deviates significantly from the rest. The transmitted signals does not contain transients, and any abnormally large signal sample is therefore noise transient. Most of the energy in a transient can be removed from the signal simply by resetting the sample to a more appropriate value, such as the mean value, for instance. An indication of the number of transients can be obtained by sorting the samples according to magnitude and then determine the decay or the number of samples above, say, 3 times the ℓ^2 norm. The downside of these easy-to-understand-and-implement ideas is that they require a considerable amount of computational power, much more than is available in a low-cost sensor.

A far less computational alternative is needed. One possible solution is to utilize the idea of transmitting in a number of channels as described in Section 4.1.1 and 4.1.2 combined with a particular design of original $\mathbf{u}_0$ and orthogonal test signals $\mathbf{u}_n$. While the overall purpose of using the test signals is to detect any kind of noise, for instance by

$$p = \frac{\langle \mathbf{y}, \mathbf{u}_0 \rangle}{\sum_{n=0}^{N} | \langle \mathbf{y}, \mathbf{u}_n \rangle |} , \tag{4.15}$$

it is possible to adapt some of them to specifically detect time-localized noise. The idea is to design some of the $\mathbf{u}$'s in a particular way, which is illustrated here with just $\mathbf{u}_1$ and $\mathbf{u}_2$. To make the notation easier the vectors in the rest of this section now represents only the samples in the chosen element, i.e. the interval in which the designed signal is

non-vanishing, and not the whole signal. First, let

$$\mathbf{u}_0 = \begin{bmatrix} \mathbf{u}_0^0 \\ \mathbf{u}_0^1 \\ \mathbf{u}_0^2 \\ \mathbf{u}_0^3 \end{bmatrix}$$

be given. Define

$$\mathbf{u}_1 = \begin{bmatrix} \mathbf{u}_1^0 \\ \mathbf{u}_1^1 \\ \mathbf{u}_1^2 \\ \mathbf{u}_1^3 \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} \mathbf{u}_1^0 & \mathbf{0} \\ \mathbf{u}_1^1 & \mathbf{0} \\ \mathbf{0} & \mathbf{u}_1^2 \\ \mathbf{0} & \mathbf{u}_1^3 \end{bmatrix}^\top \mathbf{u}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and

$$\mathbf{u}_2 = \begin{bmatrix} \mathbf{u}_2^0 \\ \mathbf{u}_2^1 \\ \mathbf{u}_2^2 \\ \mathbf{u}_2^3 \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} \mathbf{u}_2^0 & & & \\ & \mathbf{u}_2^1 & & \\ & & \mathbf{u}_2^2 & \\ & & & \mathbf{u}_2^3 \end{bmatrix}^\top \mathbf{u}_k = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad k = 0, 1 \, .$$

It is assumed that $\mathbf{u}_n^k$ have the same length for fixed k. Let $\boldsymbol{\chi}^k$ be the characteristic vector for the interval associated with $\mathbf{u}_0^k$ and with the same length as the $\mathbf{u}_0$, i.e. it is 1 on the interval corresponding to $\mathbf{u}_0^k$ and 0 elsewhere. Define then

$$p_j^k = \frac{\sum_{s=0}^{M-1} u_0[s] y[s] \chi^k[s]}{\sum_{m=0 \,\wedge\, m=j}^{J} \left| \sum_{s=0}^{M-1} u_m[s] y[s] \chi^k[s] \right|} \, ,$$

where J is the number of specially designed test signals and M is the length of $\mathbf{u}_0$. This formula does the same as (4.15) except it is confined to only parts of the signal (determined by k in $\boldsymbol{\chi}^k$). The outermost sum in the denominator does not include the $\mathbf{u}_m$ for m between 1 and the examination level, since these $\mathbf{u}$'s are not (necessarily) orthogonal to $\mathbf{u}_0$ on the target interval.

Now, suppose that $\mathbf{y}$ is the WPT of the received signal in which there is a time-localized noise occurrence in the second quarter (and suppose that there is no other noise). Then p_0^0, p_1^0, and p_2^1 is less than 1, while the remaining p's are all 1. Although this does not pinpoint the location of the transient it does reveal that something has happened in the second quarter of the signal. At the same time the fact that the remaining p's are 1 indicates that three quarters of $\mathbf{y}$ is noise-free. By changing the inner product $\langle \mathbf{y}, \mathbf{u}_0 \rangle$ from being a correlation between all samples in $\mathbf{y}$ and $\mathbf{u}_0$ to a correlation between the noise-less three quarters of $\mathbf{y}$ and the corresponding samples in $\mathbf{u}_0$ a good estimate of G can still be obtained, even without extra effort if the initial estimate was obtained as the

sum of the estimates based on each quarter. Note also that the designed $\mathbf{u}_1$ and $\mathbf{u}_2$ still complies with the requirements of Section 4.5.1. This means that the above suggested calculations does not require extra computations besides those needed for the validation described in Section 4.9 since the inner products have to be determined as a part of the validation procedure, anyway.

This method can of course be extended to include more **u** signals. The effectiveness of this approach does not increase linearly with the number of included **u** signals, however. As the signals becomes more segmented, i.e. smaller and smaller parts are orthogonal to equally small parts of the previous **u**'s, the p's becomes less accurate, and also at some point the management of the p's becomes more extensive than an exhaustive search for abnormal samples to exclude from the gain estimation.

Of course, knowing that something has happened in the second quarter of **y** provides the opportunity to concentrate a denoising attempt on a relatively small part of the signal (recall that in this section **y** in itself represents only a fraction of the transformed signal). If the noise is indeed a transient, and not a corruption of the majority of the second quarter, resetting the largest samples in the second quarter to what they should have been given the current estimate of the gain would provide a little more accurate estimate of the gain. Note that resetting samples to obtained increased accuracy of G should be done with care. As more samples are reset to expected values the signal comes closer to being equal to a scaling of $\mathbf{u}_0$ and the various p values thus increase correspondingly. At some point all the p values becomes close to 1 indicating a highly accurate estimate, although the reason for the high p values is really that the **y** signal has been adapted to fit the test signals.

Alternatively, the transients can be removed from the received signal prior to transformation. It is easy to get an approximate location of the transients, since each sample in **y** corresponds to only a few samples in the signal prior to transformation. The downside is that once a transient has been removed the signal has to be transformed again. The advantage of this more cumbersome denoising is that the transient is removed from all elements in the decomposition, and not just the element representing the chosen transmission channel. If the other elements are parts of a procedure for finding holes in the noise, as described in Section 4.5.1 and 4.7.2, this method might be preferable.

The time-localized denoising is illustrated in Section 5.2 in the next chapter, where a couple of experiments with wavelet modulated transmission signals are presented.

4.8 Polynomial Decomposition

One of the challenges when reducing the hardware to a minimum is low frequency noise, because this is often removed by mechanical and electrical filters. Although these are still necessary in order to avoid saturation of the ADC, they are usually less effective with reduced costs, and it is expectable to experience some degree of low frequency noise in the digital signals. This noise contribution is in many cases of a significant amplitude compared to the transmitted signals, and some means of denoising is necessary. Not because it is low frequency noise, but because the detection is less robust when the noise

energy is much higher than the signal energy.

Fortunately, the wavelet transform is well suited to separate frequencies, and the low frequency content will therefore influence only the few lowest frequency bands (which consequently are never used for transmission). Things are not so easy with the RS sequences as the RST does not separate frequencies. This section is therefore dedicated to presenting a method for removing the low frequency energy from the received signal in the case where an RS sequence in present in the signal.

One possible method of removing this noise is obviously the use of a band pass filter, and this would probably be the easiest approach from a pure denoising point of view. However, it is important to remember that the received signal contains an RS sequence which it is desirable to leave untouched. Any spread spectrum sequence will inevitably be affected by filtering and any other denoising attempt for that matter, and the chosen denoising method must therefore have an easily predictable influence on these sequences. It is demonstrated in Chapter 12 on linear transforms applied to RS sequences that the effect of a block diagonal matrix, i.e. on the form

$$\begin{bmatrix} \mathbf{B} & & & \\ & \mathbf{B} & & \\ & & \ddots & \\ & & & \mathbf{B} \end{bmatrix}, \tag{4.16}$$

applied to the RS sequence is easy to predict.

Unfortunately, filtering is not well suited for a block diagonal structure. Therefore, as an alternative the author suggests to do a polynomial-based decomposition of the signal to separate low and high frequencies. An introduction to polynomial bases can be found in Szego [74] and Chihara [16]. Though this works only in a fairly localized interval the block diagonal structure can be utilized to do the separation of frequencies for any length signal. The details of the polynomial decomposition is described in the following sections, while the presentation and discussion of the prediction of the effect with respect to the RS sequence is postponed to Chapter 12 (partly because it is a rather mathematical and extensive discussion, partly because the ease and simple structure of the prediction is an interesting result in its own right). However, the primary result of Chapter 12 is also briefly introduced in Section 4.8.2 in an applicational manner. This is because it is indeed necessary in signals from real applications, see the next chapter, to compensate for the effect that occurs when removing low degree polynomial content from the signal (and hence from the RS sequence).

The denoising method is constructed in two steps. First the polynomial basis used to decompose a signal part into polynomials is defined and discussed (in the next section). Then the block diagonal structure and the frequency aspects are discussed (in Section 4.8.2).

4.8.1 Polynomial Bases

The polynomial basis is in the following definition given in the form of a square matrix $\mathbf{\Phi}$ where each column is a polynomial sampled equidistantly on $[-1; 1)$ and of degree corresponding to the column number.

Definition 4.1 (Matrix of Sampled Polynomials)
Define the $N \times N$ matrix $\mathbf{\Phi} = [\phi_{m,n}]$ as

$$\phi_{m,n} = \sum_{k=0}^{n} c_{n,k} \Big(\frac{2m - N}{N} \Big)^k , \quad m, n = 0, \ldots, N-1 , \tag{4.17}$$

where $\mathbf{C} = [c_{n,k}]$ is a full rank $N \times N$ lower triangular, real matrix. Let $\boldsymbol{\phi}_n$ be the columns of $\mathbf{\Phi}$, i.e.

$$\begin{bmatrix} \boldsymbol{\phi}_0 & \boldsymbol{\phi}_1 & \cdots & \boldsymbol{\phi}_{N-1} \end{bmatrix} = \mathbf{\Phi} .$$

Define also the matrix $\mathbf{\Phi}_m$ as the first m columns of $\mathbf{\Phi}$. Define further $\mathbf{\Phi}^{(k)}$ as the matrix of size $2^k \times 2^k$.

The fact that the polynomials are of increasing order (and hence that changing a matrix on this form to another matrix on the same form requires an upper triangular matrix) leads to a uniqueness result.

Lemma 4.2
Let $\mathbf{\Phi}$ be an orthogonal matrix on the form (4.17)*. Then $\mathbf{\Phi}$ is unique up to a change of signs of the columns.*

The orthogonal case is also known as the Legendre Polynomials.

Proof
Any matrix on the form (4.17) is obtained by multiplying $\mathbf{\Phi}$ from the right with an upper triangular matrix $\mathbf{U}$. The resulting matrix is orthogonal if and only if $\mathbf{U}$ is orthogonal, which in turn implies that $\mathbf{U}$ is diagonal with ± 1's on the diagonal. □

Note that this lemma applies regardlessly of the sampling interval. An example of an orthogonal matrix is the 64×64 polynomial basis matrix, which is shown in Fig. 4.6.

A polynomial decomposition of a signal is done simply by multiplying the signal with $\mathbf{\Phi}^\top$ of the appropriate dimension. The result is a series of coefficients $\mathbf{c}$ which individually depends on specific polynomials content. That is, c_j depends on the polynomial content of degree j and lower in the signal (since $\mathbf{C}$ is lower triangular).

Thus, the polynomial content of degree m of a signal can be removed by projecting it onto the vector space span$\{\boldsymbol{\phi}_{m+1}, \ldots, \boldsymbol{\phi}_{N-1}\}$, and alternatively by subtracting from the signal its projection onto the vector space span $\mathbf{\Phi}_m$. For $m \ll N$ the latter procedure is far less computational demanding.

It should be noted that although the Gram matrix of $\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{N-1}$ in the Legendre case is the identity matrix the construction of the orthogonal polynomials is numerically

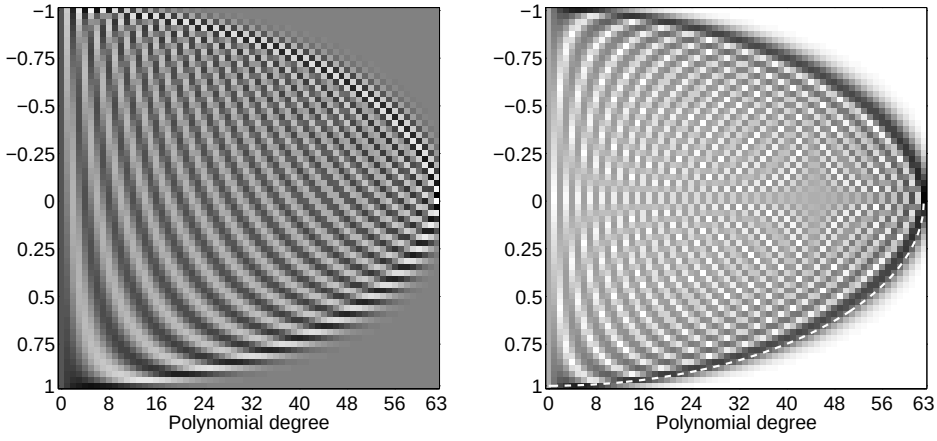


Figure 4.6: The orthogonal 64×64 matrix $\mathbf{\Phi}^{(6)}$ for some choice of column signs (left) and the absolute value of the matrix (right). The white, dashed line in the right plot shows $\sqrt{1-(p/63)^2}$ where $0 \leq p < 63$, see Conjecture 4.4.

highly unstable. The condition number of $\mathbf{\Phi}$ grows very rapidly with increasing dimensionality of the spanned space. Fortunately, the first few (low degree) vectors of the basis can be constructed without difficulty.

4.8.2 Applying Polynomial Decomposition

The idea for applying the polynomial decomposition in order to remove the low frequency noise of the signal is the following. The signal $\mathbf{x}$ is separated into a number of consecutive parts $\mathbf{x}_k$ of equal length. Each part must have length equal to some power of 2. Then the polynomial content of degree m is removed from each part. Finally, the parts are concatenated to produce the denoised signal. Mathematically this can be achieved by a multiplication by a matrix on the form (4.16). In practice this is accomplished as described in the previous section, i.e. by first determining $\mathbf{c} = \mathbf{\Phi}_m^\top \mathbf{x}_k$ followed by subtraction from the signal $\mathbf{x}_{\text{denoised}} = \mathbf{x}_k - \mathbf{\Phi}_m \mathbf{c}$.

The number of signal parts and the degree of polynomial removal is based on the sample rate and the frequencies which is to be removed, and will therefore be determined by the circumstance of the individual applications (see Section 5.4 for examples of applications of this method).

One can get an idea of frequency interpretation of this method by looking at the frequency content of the basis elements in the polynomial decomposition, that is the columns of $\mathbf{\Phi}$. This is shown in Fig. 4.7. It is clear from this figure that removing low degree polynomial content very effectively removes the low frequency content of the signal. The price paid for using this approach is also clear; some of the higher frequency content is

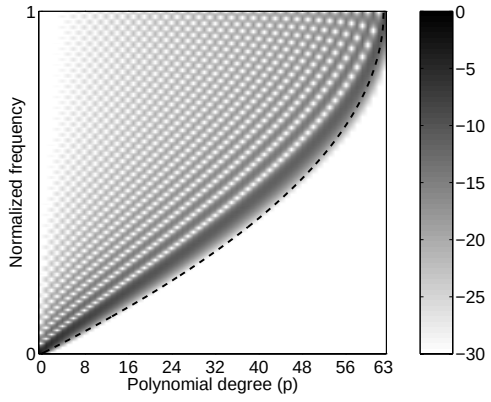


Figure 4.7: The frequency content of each column of $\mathbf{\Phi}^{(6)}$, i.e. of each polynomial in the basis. The black, dashed line shows $1 - \sqrt{1 - p/63}$ where $0 \leq p < 63$, see Conjecture 4.4. The color scale is in relative dB.

removed as well. However, it turns out that this is not a major problem in real applications.

It was mentioned in the beginning of this section that it is easy to predict the impact of removing low degree polynomial content from an RS sequence. Fortunately, it is also easy to compensate for this impact, as the following example will show. Let $\mathbf{x}$ be a signal which has been received under such circumstances that a major part of the signal energy is concentrated in the low frequencies, and a minor part of the energy is in an RS sequence. Applying the low degree polynomial removal will indeed remove most of the low frequency energy from the signal. At the same time this process alters the RS sequence such that a subsequent RST will not yield energy in only one sample, but rather in a number of samples (how to determine which ones are discussed in Chapter 12). For instance, a length 64 RS sequences, being the RST of $\mathbf{e}_0$, a vector with all but the first entry vanishing, is subjected to a third degree polynomial removal on 8 length 8 signal parts. That is, the first 8 samples have the third degree polynomial content removed, the next 8 samples are subjected to this also, and so on. Then the signal is transformed 'back'. Without removing the polynomial content the result would be a signal vanishing in all but the first entry. However, with the polynomial content removed the result is the signal shown in Fig. 4.8. Note that as long as the receiver is linear the noise is of no concern to this analysis, since the polynomial removal is a linear operation, and thus influences the RS sequence independently of the present noise.

Since the only unknown parameter in a real application is the amplitude of the RS sequence in the received signal it is possible, once this parameters has been estimated, to approximately 'undo' the effect of the polynomial removal. This is accomplished by applying to the transformed signal a method which when applied to the signal in Fig. 4.8 will yield $\mathbf{e}_0$. And this is done by multiplying the first entry in the transformed signal with the reciprocal of the first entry in Fig. 4.8, and subtract from each of the remaining entries

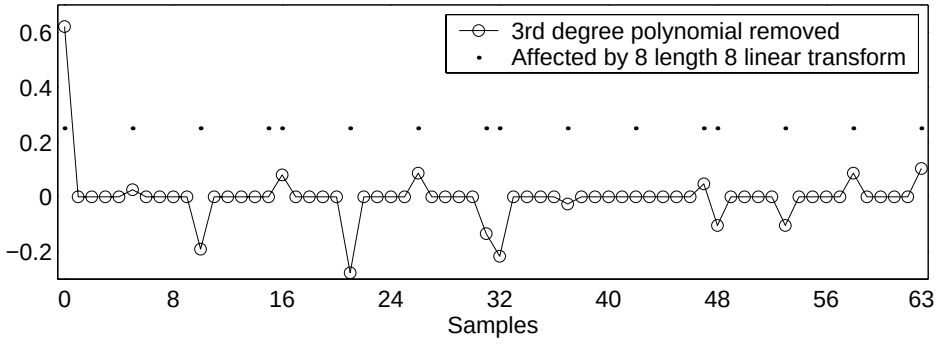


Figure 4.8: The result of removing third degree polynomial content from 8 length 8 parts of the first length 64 RS sequence (first row of matrix on the left in Fig. 4.9). The black dots mark the samples which are potentially affected by any '8 length 8' linear transform of the first length 64 RS sequence (first row of matrix on the right in Fig. 4.9).

in the transformed signal the value of the corresponding entry in the signal in Fig. 4.8.

An alternative to undoing the effect is to ignore the affected entries in the transformed signal. After all, a majority of the entries are unaffected. The black dots in Fig. 4.8 show which entries are potentially affected by any block diagonal linear transformation applied to 8 length 8 signal parts prior to the RST (this is elaborated in Chapter 12).

The polynomial removal procedure can also be described in linear algebra terms. Let $\mathcal{L}^{(3)}$ be a $2^3 \times 2^3$ matrix which projects a length 8 vector onto the space spanned by sampled polynomials of degree 4 through 7, i.e. $\mathcal{L}^{(3)} = \mathbf{I} - \mathbf{\Phi}_4^{(3)} (\mathbf{\Phi}_4^{(3)})^\top$. In order to apply this matrix to the 8 length 8 parts of the signal, define the $2^6 \times 2^6$ matrix $\mathcal{L}^{(6,3)} = \mathbf{I}_{8\times 8} \otimes \mathcal{L}^{(3)}$, where $\otimes$ is the Kronecker product, which when applied to the received signal will remove third degree polynomial content as described above. The entire process is then as follows. The transmitted signal is given by $\mathbf{P}^{(6)}\mathbf{e}_0$. When received the signal is subjected to $\mathcal{L}^{(6,3)}$ and then transformed with $\mathbf{P}^{(6)}$. The resulting signal is $\mathbf{y} = \mathbf{P}^{(6)}\mathcal{L}^{(6,3)}\mathbf{P}^{(6)}\mathbf{e}_0$, that is the first row of $\mathbf{P}^{(6)}\mathcal{L}^{(6,3)}\mathbf{P}^{(6)}$. This row is the signal shown in Fig. 4.8. The entire matrix is shown on the left in Fig. 4.9. On the right in the same figure is a matrix showing which entries (marked with black) are potentially affected by any linear transform on the form $\mathcal{L}^{(6,3)}$, i.e. when applied to 8 length 8 signal parts. This matrix is presented in Chapter 12.

When applying the polynomial removal procedure to consecutive signal parts there is a potential risk of introducing discontinuities in the signal. This happens if the edge of the polynomial approximation in one signal part matches poorly with the corresponding edge in polynomial approximation in an adjacent signal part. This mismatch occurs when the polynomials are poor approximations of the individual signal parts, and a poor approximation is the result when the signal part exhibits non-differential behavior, and if

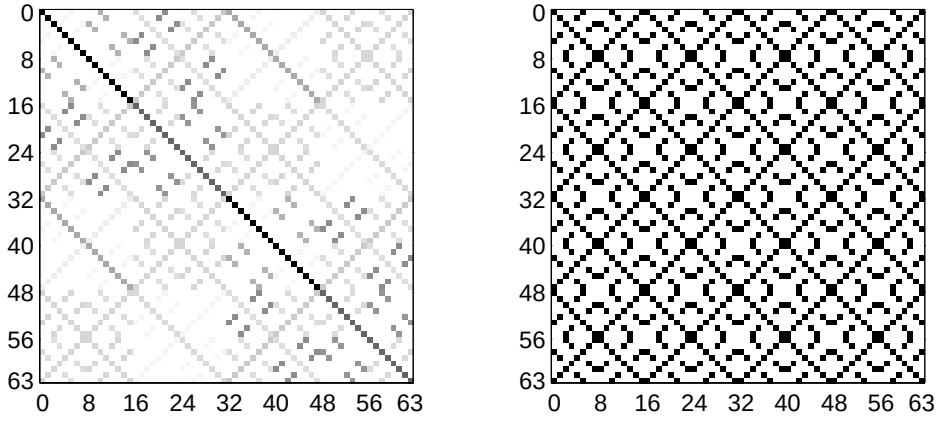


Figure 4.9: The matrix $\mathbf{P}^{(6)}\mathcal{L}^{(6,3)}\mathbf{P}^{(6)}$ (left) and the matrix indicating the entries which are potentially affected by a block diagonal linear transform applied to 8 length 8 signal parts prior to RS transformation (right).

the signal part contains a significant energy at a frequency not 'covered' by the chosen polynomial degree. The former is typically a result of transients in or saturation of the signal, while the latter happens when the disturbances oscillates too fast compared to the chosen polynomial degree and number of signal parts.

The major problem with discontinuities in the signal is that they introduce non-transmission-signal energy which degrades the subsequent estimate of the channel gain. It should be noted that the problem is not that it becomes more difficult to undo the effect of polynomial removal (as this step is independent of the quality of the approximation), but rather that the introduced energy affects all the samples after transformation, including the samples holding information on the channel gain.

4.8.3 Some Final Remarks

There seem to be an interesting relation between the two matrices presented in the previous section, i.e. the polynomial basis matrix $\mathbf{\Phi}$ and symmetric Rudin-Shapiro matrix $\mathbf{P}^{(N)}$. This relation has not been verified by a proof, and is thus given as a conjecture.

Conjecture 4.3
Let $\mathbf{\Phi}$ be an orthogonal matrix and define the $2^N \times 2^N$ matrix $\mathbf{B} = \mathbf{\Phi}^\top \mathbf{P}^N$. Then the distribution of the entries in each column of $\mathbf{B}$, or, alternatively, the distribution of the entries of $\mathbf{B}$, converges to the normal distribution with zero mean and unit variance for $N \to \infty$.

Whether this relation serves any purpose remains an open question. It is not explored in this thesis.

Before finishing the subject of polynomial filtering there are two interesting observations to make. They do not have any particular bearing on this thesis, but they hardly escape ones attention when looking at the figures 4.6 and 4.7. The functions given in this conjecture have been marked in the two figures by a white and black dashed line, respectively. The author does not have any theory or references to any theory to support these observations, and therefore they are presented here as conjectures. Having no relevance for the thesis the subject will not be pursued any further.

Conjecture 4.4
Let $\mathbf{\Phi}$ be a $N \times N$ matrix as defined in Definition 4.1.

1. *Define for fixed N the function*

$$\eta(t) = \frac{2}{N} \underset{0 \le m < N/2}{\arg\max} |\phi_{m,\lfloor tN \rfloor}|, \quad t \in [1/N; 1) \ .$$

   Then

$$\lim_{N \to \infty} \eta(t) = \sqrt{1 - t^2},$$

2. *Define for fixed N the function*

$$\hat{\eta}(t) = \frac{2}{N} \underset{0 \le k < N/2}{\arg\max} \left| \sum_{m=0}^{N-1} \phi_{m,\lfloor tN \rfloor} e^{-ikm\pi/N} \right|, \quad t \in [1/N; 1) \ .$$

   Then

$$\lim_{N \to \infty} \hat{\eta}(t) = 1 - \sqrt{1 - t}.$$

The author does not have any plans to investigate the presented conjectures any further, and interested readers are invited to attempt to verify them.

4.9 Validation of Measurements

When the measurement of the channel gain including the various forms of denoising have been completed, an estimate of the channel gain and a sequence of pure-noise samples are available. This information can be exploited in a number of ways to validate the CGM, i.e. determine whether the inaccuracy on the CGM is within acceptable limits. The following subsections presents three different methods of various mathematical complexity.

Validating a measurement means to decide whether it is useful or not useful. The common aim of the validation methods is to make this decision with a predetermined error rate. Making a error means making the wrong decision. The error rate might be set once and for all, it might be adjustable at the decision or functionality level, or it might be given by user input. In any case, it is important to distinguish between the two basic types of error, false positive (FP), i.e. deciding that a useless measurement is useful, and false negative (FN), i.e. deciding that a useful measurement is useless.

4.9.1 Threshold on the Measurements

The easiest way of validating a measurement is to fix a threshold above which a measurement is considered useful and below which it is useless. In most proximity sensors this validation is also the output of the sensor; a sufficiently large measurement means that it is most likely generated by an object close to the sensor. Typically, the threshold is set to a level which empirically yields a predetermined probability of FP under given conditions. If the distribution and variance of the noise is known it is easy to determine the threshold. This method employs a fixed threshold and is therefore a widely used method in analog sensors.

Although applying an FP-based threshold directly to the measurements is straightforward and simple to do, this approach does not yield the true error rate since it does not include the FN probability. The challenge when including the FN probability is that it depends on the noise level as well as the desired maximum sensitivity of the sensor whereas the FP probability depends on the noise level only. This is illustrate in the following example.

Assume that the noise in a given proximity sensor is normally distributed with zero mean and standard deviation 10, and that the error rate is specified to 10^{-6}. This corresponds to 4.75 times the standard deviation, and the threshold should be set to 48 accordingly (assuming that only integers are allowed). This ensures that the probability of detecting an object when there is none is 10^{-6}. However, if an object is present and results in a true CGM of 48 half of the measurements will be below the threshold, and thus the FN probability is 0.5. When the object moves closer and the true CGM increases to 96 the FN probability drops to 10^{-6}. In both cases the FP probability is 10^{-6}. If the sensor is specified to have an error rate of (at most) 10^{-6} this is only valid for the presence of objects which causes a true CGM of (at least) 96.

Note that while the fixed threshold on the measurements works fine in a white noise scenario (when the above consideration are taken into account) the method lacks the ability to properly distinguish between large measurement caused by an powerful transmission and by noise.

4.9.2 Adaptive Validation

The deficiency of the fixed FP-based threshold validation demonstrates the need for an adaptive validation method. In the following subsections two adaptive methods are reported. They are both based on the principle of regular SNR,

$$10\log_{10}\frac{y_k^2}{\sum_{n=M}^{N-1} y_n^2} \quad k = 0, 1, \ldots, M-1 \ ,$$

where M is the number of emitters and N the number of samples in the emitted sequences. This means that instead of validating the measurements by a threshold on the measurements, the validation is based on some sort of SNR, called a validation function. The

functions in the two validation methods reported here are

$$\Theta(\mathbf{y}) \equiv \frac{y_0^2}{\sum_{k=1}^{N-1} y_k^2} \tag{4.18}$$

and

$$\tilde{\Theta}(\mathbf{y}) \equiv \frac{y_0^2}{\sum_{n=1}^{N-1} y_n^2 + \beta |y_0|^3} \ , \tag{4.19}$$

respectively. This implicitly assumes that y_0 is the gain measurement and the channels y_1 through y_{N-1} are noise. Having information about the noise, it is natural to compare the (potential) signal to the noise directly as in (4.18). The other function (4.19) requires a little motivation. The idea is here that experience shows that it is sometimes necessary to introduce a mechanism for handling large outliers in time domain (transients), which is achieved for instance as in (4.19) by introduction of the cubic term in the denominator. In the sequel, design procedures will be proposed for either function. Thus, for (4.19) the challenge is to keep the detection criterion well-balanced for ordinary white noise at the same time as being able to reject transients and other time-localized disturbances.

The validation is a two step procedure. First the measurement must be above a certain threshold S, as in the previously described method. Then the validation function must be above a certain threshold α. That is, the test of the hypothesis that a gain measurement is useful is on the form

$$\mathcal{T}(S,\alpha) = \begin{cases} \text{false} & y_0 < S \ , \\ \text{false} & \Theta(\mathbf{y}) < \alpha \ , \\ \text{true} & \text{otherwise.} \end{cases}$$

The same applies to $\tilde{\mathcal{T}}$ and $\tilde{\Theta}$. The validation test that uses a threshold on the signal level is not be included in this work.

The purpose of using these two methods is to be able to properly validate measurements in the case of severe noise. At the same time they must be able to provide a validation with a predetermined error rate for normally distributed noise. This goes for FP as well as FN errors. The ability of the validation methods to detect non-random noise is demonstrated in the next chapter and will not be discussed any further here. The remaining part of this section is dedicated to determining the parameters S, α, and β.

4.9.3 First Validation Method

Two parameters have to be determined in order to use the first validation method. They can obviously be determined empirically by trail and error. However, if one wants to quantify the error rates it is necessary to know the relation between instances of the (almost) stochastic process $\mathbf{y}$, and S and α. As argued in Section 4.9.1 it is reasonable to require the FP and FN error rates to be equal in the random-noise worst-case scenario, i.e. in the case where the signal is the weakest possible and yet still useful.

In the following a statistical model for balancing the probability of a FP decision and a FN decision is presented. To reduce the complexity S is assumed to be $-\infty$, i.e. it is not a part of the validation. Therefore the entire exercise is about determining the correct α. Obviously, the probability P_{FP} of detecting a signal when none was received decreases with large values of α. And vice versa, when α is small, the probability P_{FN} of ignoring a signal that was actually received is also small. Thus, choosing α can be seen as a compromise between FP and FN risks.

The purpose of the statistical model is to determine the optimal threshold α. We define an optimal threshold as that value of α for which the probability for a worst-case FN decision based on $\mathcal{T}(\alpha)$ equals the probability for an FP decision based on $\mathcal{T}(\alpha)$. A worst-case FN decision is understood as an FN decision in the presence of the faintest received signal which is to be considered useful. It is straightforward to modify the approach below in order to meet this compromise with a preference to either a low FP or a low FN probability.

Note that in the following model Θ has been divided by $N-1$ since this makes the denominator resemble the variance of the signal.

The statistical model and the accompanying derivations and computations are due to Jakob Stoustrup. The result presented here have also been submitted for publication elsewhere, see la Cour-Harbo and Stoustrup [51].

4.9.4 Statistical Model for Deterministic Gain Signal

The reader is reminded that if $N-1$ stochastic variables $\{Y_k\}_{k=1\ldots N-1}$ are normally distributed, $Y_k \in N(0,1)$, then $\sum_{k=1}^{N-1} Y_k^2$ belongs to the $\chi^2(N-1)$ distribution, which is a special case of the Γ distribution, $\chi^2(N-1) = \Gamma\left(\frac{N-1}{2}, 2\right)$. This means that the probability of an FP decision is

$$P_{\mathrm{FP}}(\alpha) = P(Z_0 > \alpha \bar{Z})$$

where

$$Z_0 = Y_0^2 \in \Gamma\left(\frac{1}{2}, 2\right)$$

and

$$\bar{Z} = \frac{1}{N-1}\sum_{k=1}^{N-1} Y_k^2 \in \Gamma\left(\frac{N-1}{2}, 2\right),$$

and the probability of an FN decision is

$$P_{\mathrm{FN}}(\alpha, R_{\max}) = P(R_{\max}\sigma^2 < \alpha \bar{Z}),$$

where σ^2 is the true (and unknown) variance of the noise. $R_{\max}$ is the worst-case SNR, i.e. $R_{\max} = y_{\mathrm{low}}/\sigma$, where y_{low} is the lowest detectable signal level of y_0. Note that this makes $R_{\max}$ the 'real' SNR since y_0 is (for the time being) assumed to be deterministic.

$R_{\max}$ would typically be a design parameter or adjustable by the user of the sensor. Now, the probability distribution function P_{FP} can be computed by

$$\begin{aligned} P_{\mathrm{FP}}(\alpha) &= \iint_A f_{z_0} f_{\bar{z}}\, dz_0 d\bar{z}, \;\; A = \{(z_0, \bar{z}) : z_0, \bar{z}, z_0 - \alpha\bar{z} > 0\} \\ &= \iint_A \frac{z_0^{-\frac{1}{2}} e^{-\frac{z_0}{2}}}{\Gamma(\frac{1}{2})\sqrt{2}} \times \frac{\bar{z}^{-\frac{N-3}{2}} e^{-\frac{\bar{z}}{2}}}{\Gamma(\frac{N-1}{2}) 2^{\frac{N-1}{2}}}\, dz_0 d\bar{z} \end{aligned} \tag{4.20}$$

Introducing polar coordinates, the integral in (4.20) becomes

$$\frac{2^{-\frac{N}{2}}}{\Gamma(\frac{1}{2})\Gamma(\frac{N-1}{2})} \int_0^{\arctan\frac{1}{\alpha}} \int_0^{\infty} \Big(\frac{r^{N-2} e^{-r(\cos\theta+\sin\theta)}}{\cos\theta \sin^{N-3}\theta} \Big)^{\frac{1}{2}}\, dr\, d\theta \; . \tag{4.21}$$

The double integral can be separated into two single integrals by the substitution

$$r = \frac{\bar{r}}{\cos\theta + \sin\theta} \; ,$$

that is,

$$\begin{aligned} P_{\mathrm{FP}}(\alpha) &= K_1 \int_0^{\arctan\frac{1}{\alpha}} \Big(\frac{(\cos\theta + \sin\theta)^{2-N}}{\cos\theta \sin^{N-3}\theta} \Big)^{\frac{1}{2}}\, d\theta, \\ K_1 &= \frac{2^{-\frac{N}{2}}}{\Gamma(\frac{1}{2})\Gamma(\frac{N-1}{2})} \int_0^{\infty} \bar{r}^{\frac{N-2}{2}} e^{-\frac{\bar{r}}{2}}\, d\bar{r}. \end{aligned} \tag{4.22}$$

Finally, it can be shown that the standard substitution $t = \tan\frac{\theta}{2}$ leads to the following algebraic integrand

$$P_{\mathrm{FP}}(\alpha) = 2^{\frac{N-1}{2}} K_1 \int_0^{\sqrt{1+\alpha^2}-\alpha} \frac{t^{\frac{N-3}{2}}}{(2-(t-1)^2)^{\frac{N-2}{2}}\sqrt{1-t^2}}\, dt \; . \tag{4.23}$$

Even though the integral in (4.23) is algebraic, it can not be resolved analytically. However, numerical experiments show that e.g. an adaptive recursive Newton Cotes 8 panel rule performs better on (4.23) than on (4.22). The resulting probability function for $N = 14$ is shown in Fig. 4.10 ($N = 14$ is chosen to match the experimental signals in the next chapter).

Exploiting the probability function $P_{\mathrm{FP}}(\alpha)$ which it is possible to evaluate numerically by (4.23) it is straightforward to get a calibrating curve for α under the constraint that $P_{\mathrm{FP}} = P_{\mathrm{FN}}$. To that end, we start with a value of α, and numerically determine $P_{\mathrm{FP}}(\alpha)$. Then the inverse of the $\chi^2(N-1)$ distribution function applied to $P_{\mathrm{FP}}(\alpha)$ yields the ratio between $R_{\max}$ and α. This relationship is shown in the top plot of Fig. 4.11.

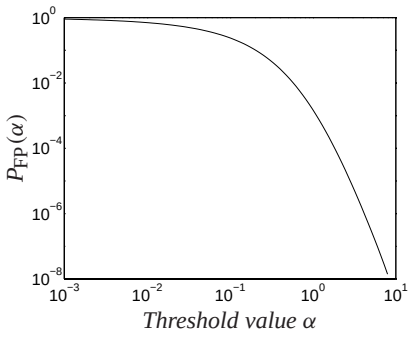


Figure 4.10: The probability for making an FP decision for a given SNR threshold α and for $N = 14$.

4.9.5 Statistical Model for Stochastic Gain Signal

In the analysis above, we have modeled the false negative situation as receiving a noise-free signal y_0 of a certain magnitude, which is incorrectly classified as noise, since the real noise signal y_1 through y_{N-1} happens to be large at the same time. This is of course unphysical to some extent, but was done in order to simplify the expressions.

A more realistic model is obtained by assuming that y_0 is an outcome of a stochastic variable, also in the false negative decision case. The analysis in principle involves the same steps as above, but the algorithm to compute the calibration curve now becomes a bit more involved.

With respect to the false positive decision case, nothing is changed and the false positive probabilities as a function of the threshold value α can be precomputed. In order to determine the calibration case, the best approach is to choose a grid of values for the threshold value, α. Then the task is to determine for each value of α, a corresponding value of the signal-to-noise-ratio (SNR) which leads to the same probability for a false negative decision as for a false positive decision for that α. It is obvious that for a fixed value of α, the false negative probabilities are monotone (non-decreasing) functions of the SNR. Thus, the right values of SNR can be found for instance by a simple bisection approach with SNR as the independent variable. For fixed values of α and SNR, the false negative probabilities can be computed as

$$P_{\mathrm{FN}}(\alpha, \mathrm{SNR}) = P\left\{ \frac{y_0^2}{\sum_{n=1}^{N-1} y_n^2} < \alpha : y_0 \in N\left(y_{\min}, \sigma\right), y_i \in N(0, \sigma),\ i = 1 \ldots N-1 \right\}$$

where $y_{\min}$ denotes the smallest possible (mean) signal received (at least the smallest for which the algorithm is guaranteed to fulfill the specified probabilities). Introducing $\mathrm{SNR} = \frac{y_{\min}}{\sigma}$, $\xi_0 = \frac{y_0}{\sigma}$, and $\bar{\xi} = \frac{1}{\sigma^2}\sum_{n=1}^{N-1} y_n^2$ (which is then $\chi^2(N-1)$ distributed), we

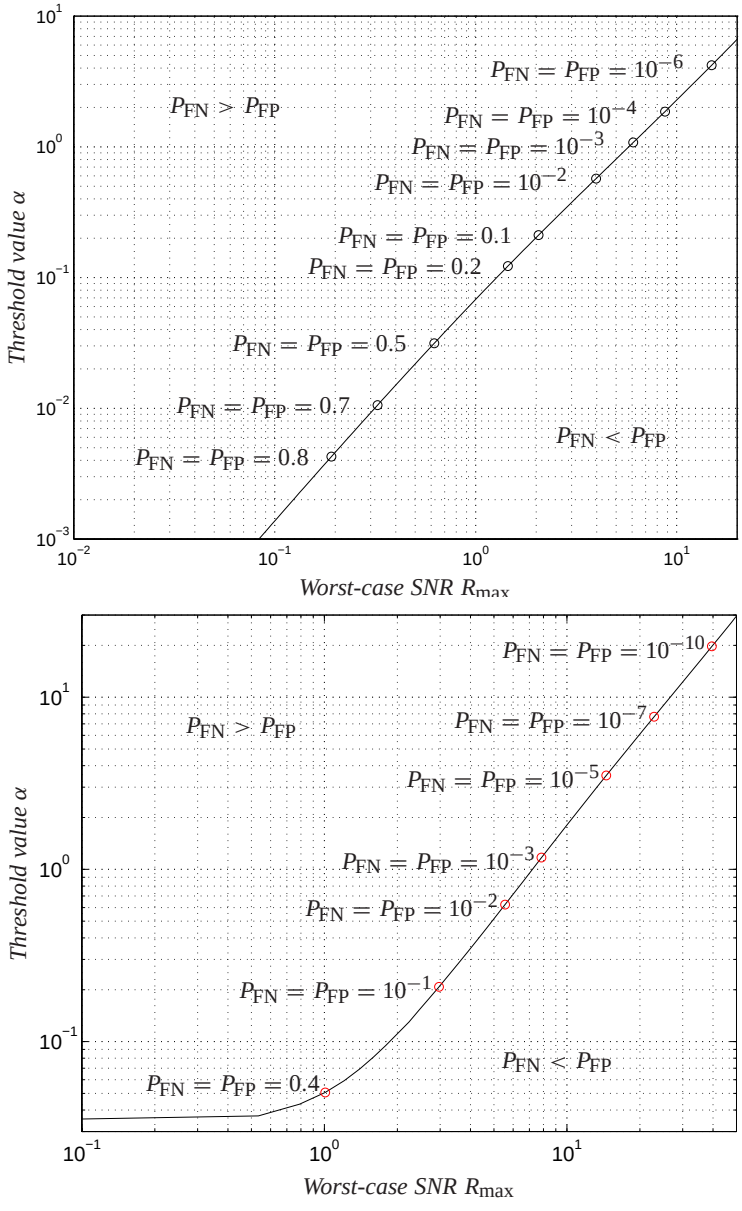


Figure 4.11: The curves show the relationship between the worst-case SNR value $R_{\max}$ and the threshold value α in the test $\mathcal{T}(\alpha)$. The top plot for a deterministic y_0 and the bottom plot for a stochastic y_0. Here $N = 14$.

obtain

$$\begin{aligned} P_{\text{FN}}(\alpha, \text{SNR}) &= P\left\{\frac{\sigma^2\xi_0^2}{\sigma^2\bar{\xi}} < \alpha : \xi_0 \in N(\text{SNR}, 1), \bar{\xi} \in \chi^2(N-1)\right\} \\ &= P\left\{\xi_0^2 < \alpha\bar{\xi} : \xi_0 \in N(\text{SNR}, 1), \bar{\xi} \in \chi^2(N-1)\right\} \\ &= \iint_{\xi_0^2 < \alpha\bar{\xi}} f_{N(\text{SNR},1)}(\xi_0)\, f_{\chi^2(N-1)}(\bar{\xi})\, d\xi_0 d\bar{\xi} \\ &= \int_{\bar{\xi}=0}^{\infty} \left(F_{N(\text{SNR},1)}\left(\sqrt{\alpha\bar{\xi}}\right) - F_{N(\text{SNR},1)}\left(-\sqrt{\alpha\bar{\xi}}\right)\right) f_{\chi^2(N-1)}(\bar{\xi})\, d\bar{\xi} \end{aligned} \tag{4.24}$$

The calibration curve resulting from applying the bisection algorithm mentioned above based on numerical evaluations of (4.24), can be seen in the bottom plot of Fig. 4.11. Comparing the two subplots of Fig. 4.11, it is easy to see that they are almost identical for large values of SNR, whereas the bottom plot suggests larger values of α for small SNR values. In fact, the bottom curve has a left, horizontal asymptote for SNR $\to -\infty$, which is easy to verify. Indeed, as SNR $\to -\infty$ the measurements y_0 for the false positive and the false negative decisions asymptotically belong to the same distribution, i.e. $N(0, 1)$. Thus, the requirement $P_{\text{FP}} = P_{\text{FN}}$ in the limit leads to

$$\begin{aligned} &P\left\{\xi_0^2 < \alpha\bar{\xi} : \xi_0 \in N(0, 1), \bar{\xi} \in \chi^2(N-1)\right\} \\ &\qquad = P\left\{\xi_0^2 > \alpha\bar{\xi} : \xi_0 \in N(0, 1), \bar{\xi} \in \chi^2(N-1)\right\} \end{aligned} \tag{4.25}$$

However, since these two probabilities in this case are obviously related also by $P_{\text{FP}} = 1 - P_{\text{FN}}$, we obtain $P_{\text{FP}} = P_{\text{FN}} = \frac{1}{2}$, which corresponds to a unique value of α.

While the P_{FP} is only dependent on α, and decreases with increasing α, the probability of making a FN error is also dependent on the signal level (as described previously). Thus, the P_{FN} curve can be plotted for fixed y_{low}, i.e. fixed SNR. In Fig. 4.12 seven such curves are plotted for the SNR values corresponding to the seven marked points on the $P_{\text{FP}} = P_{\text{FN}}$ curve in the bottom plot of Fig. 4.11. As expected P_{FN} increases with increasing α. The points where the P_{FP} curve intersects with the seven P_{FN} curves correspond to the points marked in the bottom plot of Fig. 4.11.

4.9.6 Second Validation Method

The first validation method performs well in most cases. This is demonstrated in the second test setup in Section 5.3 in the next chapter. However, it is also demonstrated that particularly powerful noise burst have a tendency to be accepted as useful measurements. Experiments have shown that introducing the third order term in the validation function helps to prevent this. Unfortunately, this also makes the choice of parameters more complicated. This subsection first shows the meaning of the various parameters and give some

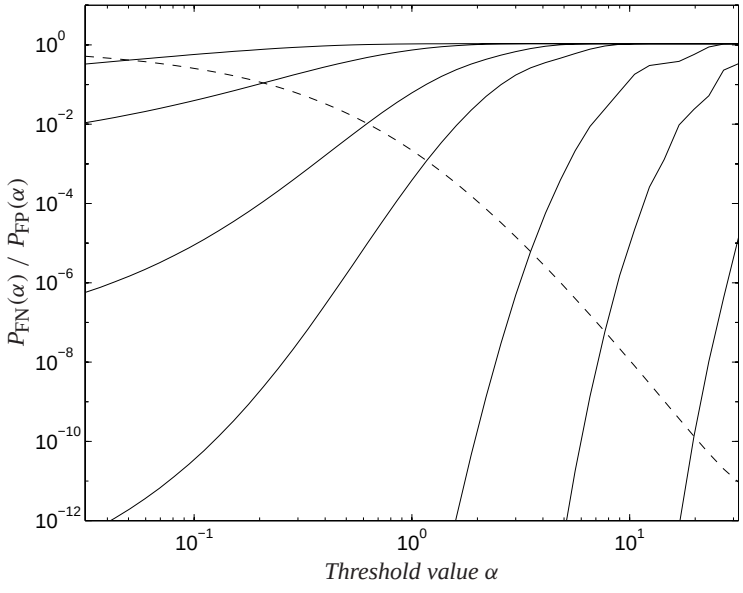


Figure 4.12: The one P_{FP} curve (dashed) and the P_{FN} curves (solid) corresponding to the marked points in the top plot of Fig. 4.11. The crossing points gives the α value which gives $P_{\mathrm{FP}} = P_{\mathrm{FN}}$ for the SNR used for drawing each of the seven P_{FN} curves. The slight irregularity of some of the curves are due to numerical instability.

hints on how to choose them. Subsequently, a statistical analysis is performed in order to calculate the $P_{\mathrm{FP}} = P_{\mathrm{FN}}$ calibration curve for the sensor system.

4.9.7 Simple Design Rules for the Second Validation Method

An obvious concern with the introduction of the third order term is that for fixed noise energy $\mathcal{T}(\alpha)$ evaluates to false for sufficiently large y. In practice this is to be handled by choosing β such that this does not occur within the dynamical range of y. To be able to do that it is necessary to understand how $\mathbf{y}$, α, and β influence the $\tilde{\Theta}$ function.

In contrast to Θ in the first validation method $\tilde{\Theta}$ is not monotonically increasing for fixed noise level since

$$\tilde{\Theta}(0, \Lambda) = 0 \qquad \text{and} \qquad \lim_{y \to \pm\infty} \tilde{\Theta}(y, \Lambda) = 0 \,,$$

where

$$\tilde{\Theta}(y_0, \Lambda) \equiv \tilde{\Theta}(\mathbf{y}) \quad \text{with} \quad \Lambda = \sum_{n=1}^{N-1} y_n^2 \,.$$

This means that the test $\tilde{\mathcal{T}}(\alpha)$ only evaluates to true inside some interval $(y_{\min}; y_{\max})$ (and its negative counterpart since $\tilde{\Theta}$ is an even function). Although y in a real signal can be negative this only happens when the noise has significantly more energy than the signal, in which case the measurement is considered invalid without testing. Thus, only the positive solutions are of interest. These are

$$y_{\min} = \frac{4K_1 - K_1^2 - 4 - i\sqrt{3}(K_1^2 - 4)}{12K_1\alpha\beta}$$
$$y_{\max} = \frac{K_1^2 + 2K_1 + 4}{6K_1\alpha\beta}$$

where

$$K_1 = \left(8 + 12\beta\sqrt{81\alpha^3\beta^2\Lambda^2 - 12\Lambda} - 108\alpha^3\beta^2\Lambda\right)^{1/3} \,.$$

Note that the solutions are indeed real, although the imaginary unit is present in the formula. This is possible because K_1 is complex. Since $\tilde{\Theta}$ is continuous and $\tilde{\Theta}(0, \Lambda) = \tilde{\Theta}(\infty, \Lambda) = 0$ it has at least one maximum for some $y > 0$. Whenever α is larger than this maximum $y_{\min}$ and $y_{\max}$ does not evaluate to a real value. The maximum is found by means of the derivative, i.e.

$$\frac{d}{dy}\tilde{\Theta}(y, \Lambda) = \frac{2y}{\Lambda + \beta y^3} - \frac{3y^4\beta}{(\Lambda + \beta y^3)^2} \,.$$

The equation $\tilde{\Theta}'(y, \Lambda) = 0$ reduces to a third degree monomial equaling a non-zero constant, and thus has two complex and one real solution, the latter being

$$y_{\mathrm{top}} = \mathrm{argmax}_{y>0}\, \tilde{\Theta}(y, \Lambda) = \left(\frac{2\Lambda}{\beta}\right)^{1/3}$$

and the maximum value is

$$\tilde{\Theta}_{\max} = \tilde{\Theta}(y_{\text{top}}, \Lambda) = \frac{1}{3}\Big(\frac{2}{\beta\sqrt{\Lambda}}\Big)^{2/3}.$$

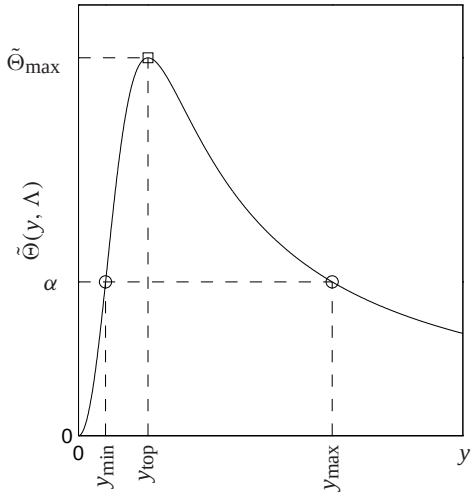


Figure 4.13: The meaning of the variables defined in Section 4.9.6.

All the variables defined above are shown in Fig. 4.13, and some examples of $\tilde{\Theta}$ curves for various values of Λ and β are shown in Fig. 4.14.

Notice that changing Λ (as in the left plot) has an almost negligible effect on $y_{\max}$ when $\tilde{\Theta}_{\max}$ is somewhat larger than α, while the same is true for the relation between $y_{\min}$ and β. The reason for this effect follows immediately from the validation function since (for $y \geq 0$)

$$\frac{y^2}{\Lambda + \beta y^3} = \alpha \quad \approx \quad \begin{cases} y = (\alpha\beta)^{-1} & \text{for } y \gg y_{\text{top}} \\ y = \sqrt{\alpha\Lambda} & \text{for } y \ll y_{\text{top}} \end{cases} \tag{4.26}$$

Actually, y does not have to be very far from y_{top} for the approximations to be fairly accurate since y^3 either becomes dominant or vanishes rapidly for increase or decreasing y, respectively.

The main purpose of (4.26) is not to provide an approximation of the validation function for implementation purposes, but to provide a tool for determining the parameters α and β in a real application. They can namely be chosen (crudely, at least) in the following way. Since Λ is often known approximately, either as a rough estimate, as a range of typical values, or statistically, α can be chosen according to the desired FN and FP rate (i.e. probability of FN and FP). Subsequently, β is chosen such that $(\alpha\beta)^{-1}$ is approximately

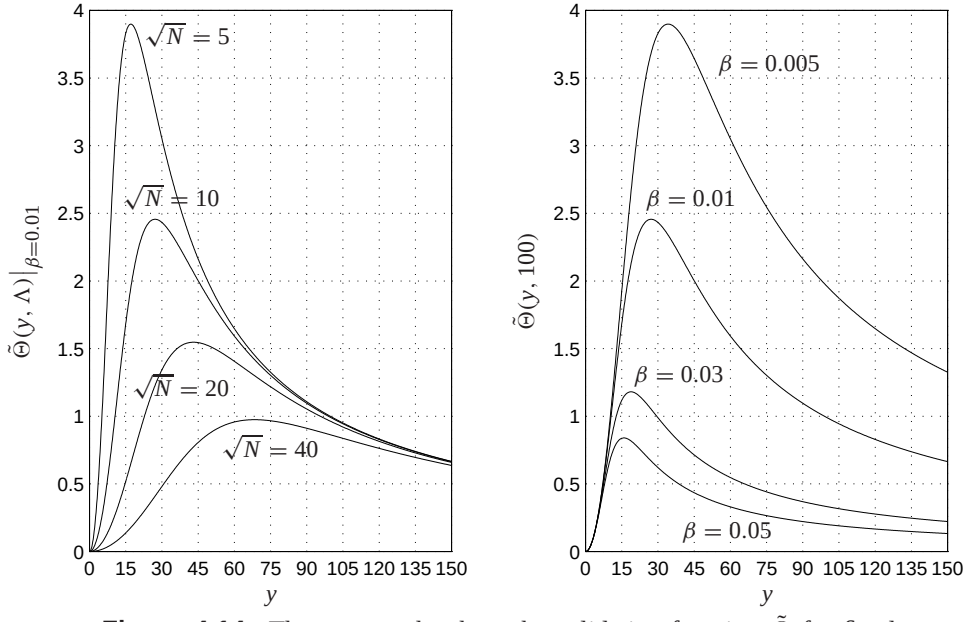


Figure 4.14: The two graphs show the validation function $\tilde{\Theta}$ for fixed $\beta = 0.01$ and varying $\sqrt{\Lambda}$ (left), and for fixed $\sqrt{\Lambda} = 10$ and varying β (right). Notice that $y_{\min} \approx \sqrt{\alpha\Lambda}$ and $y_{\max} \approx (\alpha\beta)^{-1}$ whenever $\tilde{\Theta}_{\max} \gg \alpha$.

equal to the largest attainable value of y. It is better to choose β slightly smaller rather than slightly larger since for β too large $\mathcal{T}(\alpha)$ might evaluate to false in the case where y is close to its upper limit and Λ is larger than generally expected (and that is usually undesirable). In the next chapter an example on choosing α and β is given.

4.9.8 Computing the Calibration Curve for the Second Validation Method

In order to compute the optimal threshold value α for the second validation function $\tilde{\Theta}$, we assume that a situation which is supposed to lead to a negative decision is modeled by:

$$y_0 \in N(0,\sigma)\,, \qquad y_i \in N(0,\sigma)\,,\ i = 1 \ldots N-1$$

whereas a situation which is supposed to lead to a positive decision is modeled by

$$y_0 \in N(y_{\min},\sigma)\,, \qquad y_i \in N(0,\sigma)\,,\ i = 1 \ldots N-1$$

where $y_{\min}/\sigma$ is the worst-case signal-to-noise-ratio. Introducing this in the second validation function, we obtain:

$$\begin{aligned}
P_{\mathrm{FP}}(\alpha) &= P\left\{\frac{y_0^2}{\sum_{n=1}^{N-1} y_n^2 + \beta|y_0|^3} > \alpha : y_0 \in N(0,\sigma), y_i \in N(0,\sigma)\,,\ i = 1 \ldots N-1\right\} \\
&= P\left\{\frac{\sigma^2\xi_0^2}{\sigma^2\left(\frac{1}{\sigma^2}\sum_{n=1}^{N-1} y_n^2\right) + \sigma^3\beta|\xi_0|^3} > \alpha : \xi_0 \in N(0,1),\right. \\
&\qquad\qquad \left. y_i \in N(0,\sigma)\,,\ i = 1 \ldots N-1\right\} \\
&= P\left\{\xi_0^2 > \alpha\bar{\xi} + \alpha\beta\sigma|\xi_0|^3 : \xi_0 \in N(0,1), \bar{\xi} \in \chi^2(N-1)\right\} \\
&= \int_{\xi_0=-\infty}^{\infty} \int_{\bar{\xi}=0}^{\xi_0^2\left(\frac{1}{\alpha}-\beta\sigma|\xi_0|\right)} f_{N(0,1)}(\xi_0)\, f_{\chi^2(N-1)}(\bar{\xi}) d\xi_0 d\bar{\xi} \\
&= \int_{\xi_0=-\infty}^{\infty} f_{N(0,1)}(\xi_0) F_{\chi^2(N-1)}\left(\xi_0^2\left(\alpha^{-1} - \beta\sigma|\xi_0|\right)\right) d\xi_0 d\bar{\xi} \\
&= 2\int_{\xi_0=0}^{\infty} f_{N(0,1)}(\xi_0) F_{\chi^2(N-1)}\left(\xi_0^2\left(\alpha^{-1} - \beta\sigma\xi_0\right)\right) d\xi_0 d\bar{\xi}
\end{aligned}$$

The P_{FP} curve for the second validation methods is shown in Fig. 4.15. Note that because of the third order term in the validation function the noise variance σ^2 is not only represented in $R_{\max}$, but also in the equations leading to the P_{FP} curve. The curve has been obtained by numerical integration.

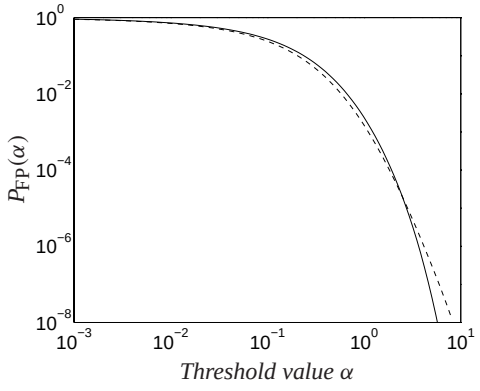


Figure 4.15: The probability for making an FP decision with the second validation method for a given threshold α, for $N = 14$, $\beta = 0.01$, and $\sigma = 3$. The dashed curve is P_{FP} for the first validation method (same as Fig. 4.10).

Similarly, for a given SNR, we can compute the risk of a FN decision. In that case:

$$
\begin{aligned}
P_{\mathrm{FN}}(\alpha, \mathrm{SNR}) &= P\left\{ \frac{y_0^2}{\sum_{n=1}^{N-1} y_n^2 + \beta |y_0|^3} < \alpha : y_0 \in N\left(y_{\min}, \sigma\right), \right. \\
&\qquad \left. y_i \in N(0,\sigma)\,,\ i = 1 \ldots N-1 \right\} \\
&= P\left\{ \frac{\sigma^2 \xi_0^2}{\sigma^2 \left(\frac{1}{\sigma^2} \sum_{n=1}^{N-1} y_n^2 \right) + \sigma^3 \beta |\xi_0|^3} < \alpha : \xi_0 \in N(\mathrm{SNR}, 1), \right. \\
&\qquad \left. y_i \in N(0,\sigma)\,,\ i = 1 \ldots N-1 \right\} \\
&= P\left\{ \xi_0^2 < \alpha \bar{\xi} + \alpha\beta\sigma |\xi_0|^3 : \xi_0 \in N(\mathrm{SNR}, 1), \bar{\xi} \in \chi^2(N-1) \right\} \\
&= 1 - \int_{\xi_0=-\infty}^{\infty} \int_{\bar{\xi}=0}^{\xi_0^2 \left(\frac{1}{\alpha} - \beta\sigma|\xi_0|\right)} f_{N(\mathrm{SNR},1)}(\xi_0)\, f_{\chi^2(N-1)}(\bar{\xi}) d\xi_0 d\bar{\xi} \\
&= 1 - \int_{\xi_0=-\infty}^{\infty} f_{N(\mathrm{SNR},1)}(\xi_0) F_{\chi^2(N-1)}\left(\xi_0^2 \left(\frac{1}{\alpha} - \beta\sigma|\xi_0| \right) \right) d\xi_0 d\bar{\xi}
\end{aligned}
$$

It is also possible to obtain the $P_{\mathrm{FP}} = P_{\mathrm{FN}}$ curve. The computation is complicated by the fact that the effect of α and SNR can not be separated in the expression for $P_{\mathrm{FN}}(\alpha, \mathrm{SNR})$. The most efficient way to obtain the curve is to fix a value of SNR. Then both the P_{FP} and the P_{FN} curves are monotone in α where the former is non-increasing

and the latter is non-decreasing, which implies that they have a unique intersection. Due to the smoothness of the two curves, e.g. an algorithm based on bisection in α converges very fast. In each step of the bisection, two numerical single integrations have to be performed. These two integrals (see above) are numerically sensitive, but can computed with some care. The result can be seen in Fig. 4.16. Note, that the calibration curve has a left, horizontal asymptote corresponding to the limiting case $P_{\mathrm{FP}} = P_{\mathrm{FN}} = \frac{1}{2}$ as the SNR goes to zero. In a doubly logarithmic plot, it also has a right asymptote, which in fact is the curve $\alpha = \frac{1}{\beta\sigma \cdot \mathrm{SNR}}$

Note also, that the $P_{\mathrm{FP}} = P_{\mathrm{FN}}$ probabilities for the calibration curve always appear in pairs at the same horizontal level. This is due to the fact that P_{FP} is uniquely given by α, and thus independent of the value of SNR.

In contrast to the first validation method the curve is upwards bounded. This is because for α sufficiently large $\mathcal{T}(\alpha)$ always evaluate to false (as explained above).

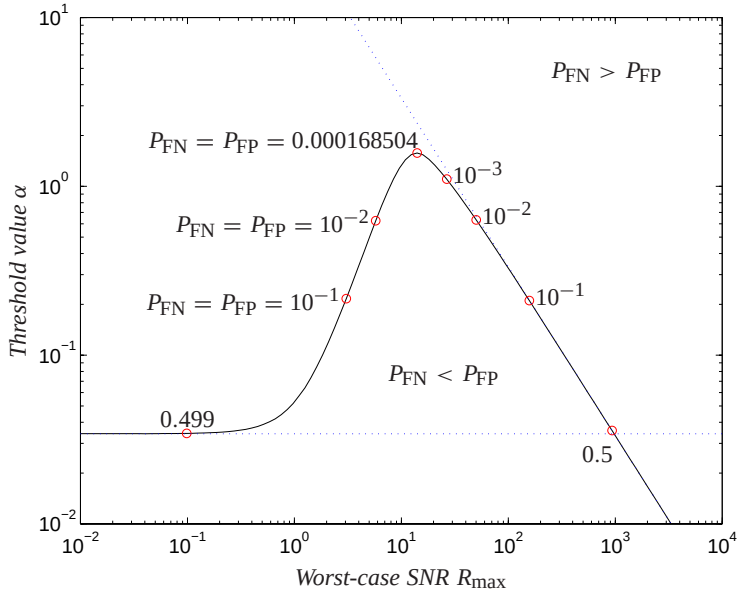


Figure 4.16: The curve shows the relationship between the worst-case SNR value of $R_{\max}$ and the threshold value α in the test $\tilde{\mathcal{T}}(\alpha)$. The dashed lines shows the left and right asymptotes. Note that there is a unique minimal value of $P_{\mathrm{FP}} = P_{\mathrm{FN}}$ at the top of the curve. The parameters are $N = 14$, $\beta = 0.01$, $\sigma = 3$.

A reasonable way to design a sensor system using the second validation method would be to balance the probabilities for $\mathrm{SNR}_{\mathrm{worst}}$ and $\mathrm{SNR}_{\mathrm{best}}$ by tuning β until the weakest and the strongest signals yield the same value of α at Figure 4.16. In that case, also the two corresponding set of probabilities will all be equal: $P_{\mathrm{FP,worst}} = P_{\mathrm{FN,worst}} = P_{\mathrm{FP,best}} =$

$P_{FN,best}$ whereas all probabilities for false decisions will be smaller in the interval in between the best-case and the worst-case.

5 Results

Having introduced a series of signal processing methods the time has come to put these methods to the test. While simulations is an invaluable tool for designing such algorithms it is only by implementation in a real setup that a method can be ultimately verified. This is indeed the purpose of this chapter.

All signals originate from setups, i.e. none of the signals presented in this chapter are synthesized (except the designed transmission signals, of course). Nor have they been altered in any way after they have been recorded.

5.1 Experimental Setups

A majority of all the methods presented in the previous chapter has been tested in an experimental setup. A total of five setups has been used, four of which are presented in this chapter. The fifth setup has been used for measuring a reflection intensity map (see Chapter 8). The primary purpose of the setups has been to develop the CGM algorithm by experimenting and testing numerous ideas. An important outcome of spending resources on physical setups (as an addition to performing simulations on a computer) is that it became clear that electrical and optical effects often play a significant role in the analog transmission of signal, that is from it leaves the signal processor and until it is back on digital form. Unexpectedly large time constants, unstable oscillating circuits, uneven optical components, ineffective optical and electrical shielding, cross talk in power supplies, unrealistic real-time requirements, and many other effects that occur in real applications can ruin even the best signal processing algorithm if no precautions are taken. Actually implementing the algorithm in real setups therefore provides an invaluable input to the designing of the algorithm.

The process of experimenting and testing ideas will not be described in this thesis. The focus is on the end result, and this chapter therefore only presents the outcome of applying the algorithm in a number of setups. This section introduces the four test setups and the data acquisition and signal processing hardware used. The following four sections then discussed each setup in details. Finally, in Section 5.6 the implementation of the algorithm in software is briefly discussed.

5.1.1 Four Test Setups

The four test setups were all designed for the specific purpose of testing the methods presented in the previous chapter. Low-cost hardware components have been used throughout the setups, and only in the fourth setup has there been an attempt to optimize all hardware parameters in the electric circuits and the mechanical construction. In all cases the setups are based on low-cost infrared technology and have been tested in a laboratory at the Department of Control Engineering at Aalborg University. The four setup are the following.

- Setup 1: Modified BeoSound Ouverture 2300
- Setup 2: Multiple emitters and receiver
- Setup 3: Distantly separated emitter and receiver
- Setup 4: Sensor of commercial standard

A list of specifications of the hardware and the tested algorithms is given in Table 5.1. A more detailed introduction to each setup is given in the following sections.

Table 5.1: Important specifications of the four experimental setups.

Specification	Setup 1	Setup 2	Setup 3	Setup 4
Transmission	Diffuse reflection	Diffuse reflection	Through-beam	Controlled, diffuse reflection
Emitter	TSHA440	SFH 405	SFH 487 P	N/A
Receiver	BPW82	BP 104 F	BP 104 F	N/A
Modulation	WP	RS	RS	RS
Sampling frequency	5 kHz	2.6 kHz	1.95 kHz	8 kHz
Signal length	512	16	64	8
Block frequency	9.77 Hz	164.7 Hz	30.5 Hz	1 kHz
Noise occurrences	Low frequency Few transients	White noise Transients	Low frequency Transients	Low frequency Transients
Denoising applied	JTF filtering Remove transient	None	Polynomial	None
Validation accuracy	2/3 channels	13 channels	63 channels	15 channels
Validation methods	Segmented test signals	Regular SNR Adapted SNR	Regular SNR Adapted SNR	Regular SNR Adapted SNR
Year	1999	2001	2002	2001

The two most important components in each of the setups are the emitter and receiver. Two different receiver and four different emitter has been used (disregarding the fourth test setup). The SHF 487 emitter has been used in the setup measuring reflection intensity maps in Chapter 8. The most important specifications are copied from the data sheets for easy reference. Although these specifications does not have a direct impact on the

tested algorithms they do give an indication of the performance requirements which can be achieved using this type of emitters and receivers. The specifications for the receivers are given in Table 5.2, and for the emitters in Table 5.3.

Table 5.2: Data for receivers.

Type	OSRAM BP 104 F	TEMIC BPW82	
Peak wavelength	950	950	nm
Area	4.84	≈ 10	mm^2
Half angle	± 60	± 65	degrees
Dark current	2	2	nA
Quantum yield (η)	0.9	N/A	$\frac{\text{electrons}}{\text{photon}}$
Raise/fall time	20	100	ns
Capacitance	48	70	pF

Table 5.3: Data for emitters.

Type	OSRAM SFH 487 P	OSRAM SFH 487	OSRAM SFH 405	TEMIC TSHA440	
Peak wavelength	880	880	950	875	nm
Half angle	± 65	± 20	± 16	± 20	degrees
Intensity (continuous)	2	20	2.5	20	mW/sr
Intensity (pulsed)	30	200	≈ 60	240	mW/sr
Raise/fall time	600	600	1000	600	ns

Since this Ph.D. study is focused on algorithms and signal processing the hardware will only be described to an extent which allows for an understanding, but not for a component-by-component reproduction. This is mainly due to the author's lack of experience with and knowledge of analog circuitry. An exception is the emitter and receiver circuits in the third test setup. The receiver is copied from an application note, and the emitter is a quite simple construction which has been designed in collaboration with two colleges. The remaining hardware has been designed by more experienced 'hardware people' under the supervision of the author.

5.1.2 Data Acquisition and Signal Processing Hardware

A PC with commercially available sampling boards have been used for generating, D/A and A/D converting, and processing the signals. In the first setup from 1999 a 80486 processor at 33 MHz has been used while a Dual Pentium II 800 MHz machine was used for the second, third, and fourth test setups. The data acquisition hardware is listed in Table 5.4.

Table 5.4: Data Acquisition Hardware.

	Setup 1	Setup 2,3,4	
	DAC, ADC	DAC	ADC
Brand	ADLink	National Instruments	National Instruments
Type	ACL-8112PG	PCI-6713	PCI-6071E
Channels	8 DA, 8 AD	8	32
Resolution	10 bit	12 bit	12 bit
Sample rate (multiplex)	100 kHz	1 MHz	1.25 MHz

5.2 First Test Setup

The Ph.D. study was initiated as a response to the desire of Bang & Olufsen to have a digital method for detecting the presence of a hand, see Section 3.4 on page 23. The hand detection property is a particular feature of the BeoSound Ouverture, and therefore a detection mechanism is currently implemented in the product. This is an analog implementation and not suitable for testing a digital approach. The emitters and receivers fitted in the Ouverture were still useful, though, and two new hardware circuits were made to connect the diodes to the PC instead of to the built-in analog detector circuit.

This setup is used for testing and experimenting with the wavelet modulation, as the Rudin-Shapiro transform were not introduced in the Ph.D. project until spring 2000.

5.2.1 Setup Specifications

The emitter and receivers diodes are all the original B&O diodes in the CD player. There are two emitters and one receiver in either side of the CD player, see Fig. 5.1. The three diodes in the right side are all connected to the sampling board in the PC. The transmission frequency is 5 kHz at a 100 % duty cycle. The transmitted signal is fixed, and generated by a three times inverse WPT of $\mathbf{u}_0$ where the basis is the eight elements on the fourth level. Prior to transmission the signals are scaled by 2048 and shifted by $+2048$ to fit the 10 bit DAC. The original signal and the transmission signal are shown in Fig. 5.2(a) and (b) (only the first half of the transmission signal is shown since the two halves are equal). The filter is Symlets 8 [27], and periodization is used to apply the WT to the finite signal, see Section 9.5.

To be able to determine the accuracy of the estimate CGMs the transmission signal is not actually emitted and received, but rather added to the received signal (where no transmission took place). The receiver has been tested and it is very close to being linear in behavior and the signal resulting from adding instead of transmitting the signal is therefore believed to be quite realistic. The 'genuine' part of the received signal is sampled with the infrared photodiode at 5 kHz, and the noise vector $\mathbf{e}_t$ is shown in Fig. 5.2c. The gain is set to $G = 0.00642$, and the constructed, received signal in shown in figure 5.2(d), and the transform of this is shown in 5.2(e). The first part of this signal is intentionally missing. This is because all the energy at DC is located here, and the mean of this first

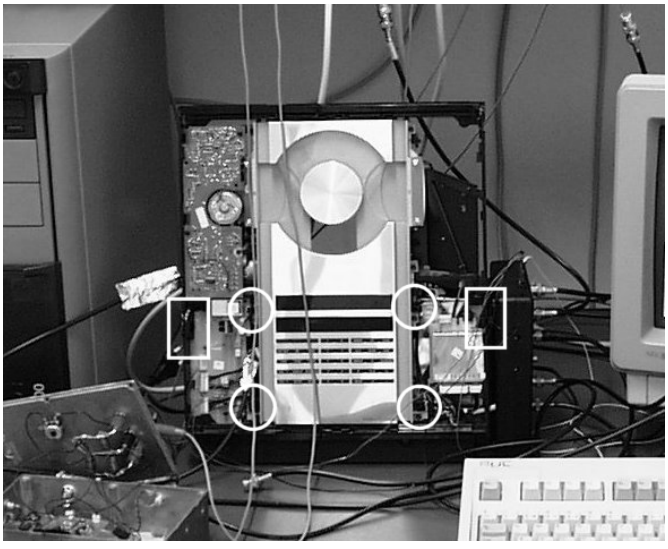


Figure 5.1: The BeoSound Ouverture with front panels removed. The four circles show the location of the emitters and the two rectangles show the location of the receivers. The black box on the right side contains the driver and amplifier circuits for the diodes.

part is several thousands meaning that this part is way off the plot.

5.2.2 Accuracy of Estimated Gain, Mean, and Variance

The appearance of the original signal in the received, transformed signal is obvious, and it seems that there is a good SNR allowing for a fairly accurate estimate of G. Since the zeroth moment of $\mathbf{u}_0$ is zero the best estimate of G is simply $\mathbf{u}_0^\top \mathbf{y}/\mathbf{u}_0^\top \mathbf{u}_0$ (under the assumption that the present noise is normally distributed). As the gain is known in this test it is possible to determine the real accuracy of G, and not just the approximated accuracy p from (4.3). The first row of Table 5.5 shows the accuracy of the estimate of G, μ, and

Table 5.5: Results of applying the least square method (4.8)–(4.10).

Description	Fig	SNR	G	$\lvert\Delta G\rvert$	$\Delta\mu$	$\Delta\sigma$
No extra disturbance	5.2(d)	-3.7 dB	0.00642	1.1%	0.000%	0.003%
		-24 dB	0.000642	11%	0.000%	0.003%
Added 200 to sample 177	5.3(a)	-8.0 dB	0.00642	13%	0.015%	62%
		-28 dB	0.000642	129%	0.015%	62%
More transients	5.4(a)	-12 dB	0.00642	34%	0.055%	164%
		-32 dB	0.000642	344%	0.055%	164%

σ, respectively. Having the signal and noise separated it is possible to determine the true

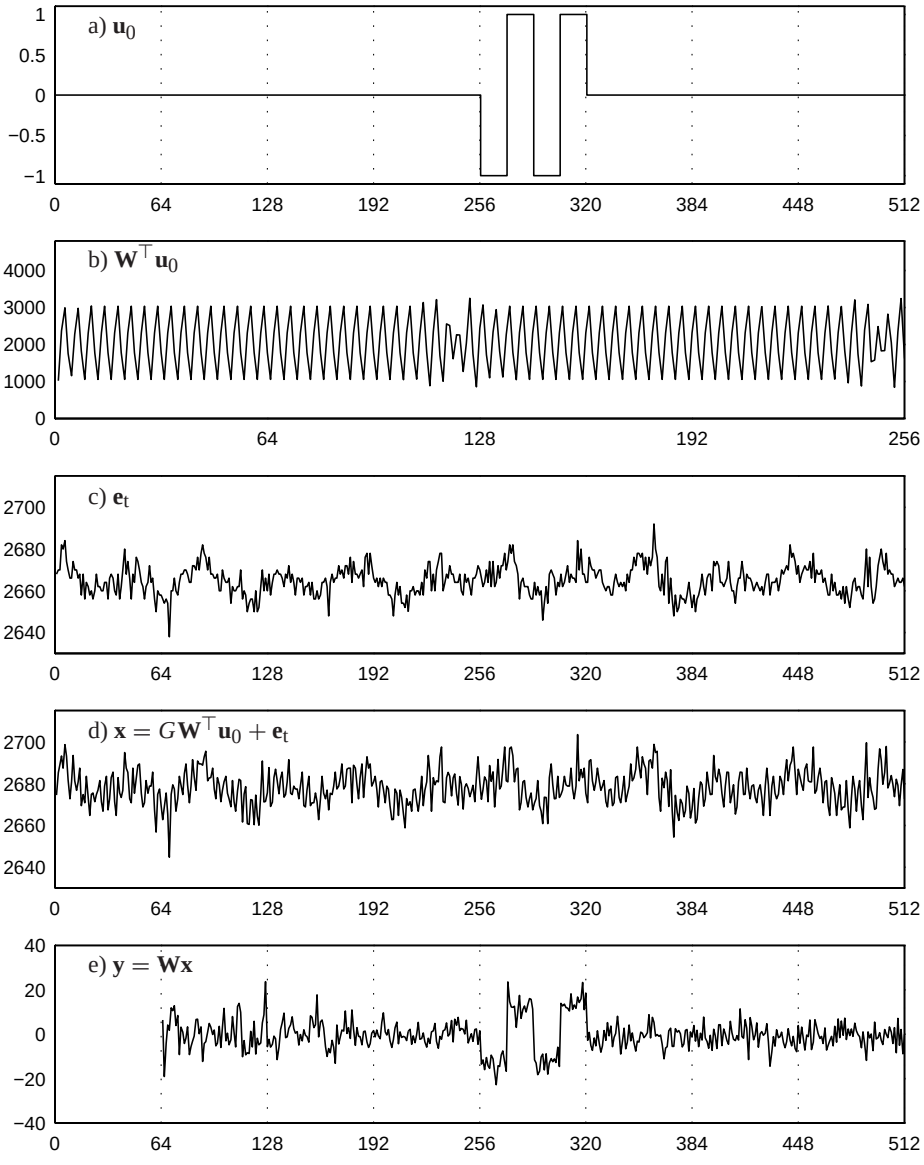


Figure 5.2: Experimental data from the first test setup. (a) The original signal $\mathbf{u}_0$, (b) first half of transmitted signal, (c) transmission noise (experimental data), (d) constructed, received signal, (e) the received signal wavelet transformed.

SNR, too. Note that this SNR is determined as

$$20\log_{10}\frac{\|G\mathbf{W}^\top\mathbf{u}_0 - E(G\mathbf{W}^\top\mathbf{u}_0)\|}{\|\mathbf{e}_t - E(\mathbf{e}_t)\|},$$

where $E(\cdot)$ is the mean value. It is obvious from the very small $\Delta\mu$ and $\Delta\sigma$ in the first two rows of Table 5.5 that the noise is indeed close to being normally distributed, and consequently, the estimated G is quite close to the true value 0.00642. The second row shows what happens if the gain is reduced by a factor 10 (the corresponding signal is not showed in any figure).

If a transient is introduced in the signal, see Fig. 5.3(a), the accuracy of the estimates decreases as shown in the third and fourth row of Table 5.5. Especially the estimated standard deviation becomes rather poor, while the mean seems surprisingly accurate. This is because the mean is quite large in the first place, and adding 200 to a single sample does not change the mean much. The estimated CGM is also less accurate. This is because the transient appears in each of the elements in the WP decomposition, and also in the fifth element, as seen on Fig. 5.3(b). This single sample is weighted by $1/64$ in the inner product, and thus increases the estimate of G correspondingly.

If more transients are introduced, as shown in Fig. 5.4(a), the accuracy of the estimates decreases. The gain is now 34% off even though the SNR has not decreased dramatically; comparing the second and fifth rows of Table 5.5 reveals that while the estimate of G is 3 times worse in the signal with transients the SNR is actually 12 dB higher. This shows that the orthogonal transform with designed signals are much more sensitive to localized noise than white noise, and thus that some sort of denoising would be appropriate.

5.2.3 Handling Transients

The presence of just a single transient in the received signal influences the channel gain to such an extent that it is worth trying to either circumvent or remove the transient. Some suggestions on how to detect and remove transients were presented in Section 4.7.4.

One method for detecting transients involves a number of the test signals $\mathbf{u}_n$. Since the original, designed signal $\mathbf{u}_0$ looks like Fig. 5.2(a) two orthogonal signals $\mathbf{u}_1$ and $\mathbf{u}_2$ could be designed like Fig. 5.3(c) and d. These two signals will help determine the accuracy of the estimate G. In Table 5.6 the inner products between y and the three designed signals are shown along with

$$p = \frac{\langle \mathbf{y}, \mathbf{u}_0 \rangle}{\sum_{n=0}^{2} |\langle \mathbf{y}, \mathbf{u}_n \rangle|}.$$

The first row shows the results of the signal with just regular transmission noise (the same as the first row of Table 5.5). The good estimate of G is indicated by a relatively high p. Remember that $p \leq 1$ always. Introducing the transient in the signal increases the inner product $\langle \mathbf{y}, \mathbf{u}_0 \rangle$. At the same time the transient also increases the amplitude of the inner product with the test signals $\mathbf{u}_1$ and $\mathbf{u}_2$. As these two inner products responds to noise only, and not to the transmitted signal, the large values in the second row indicates that

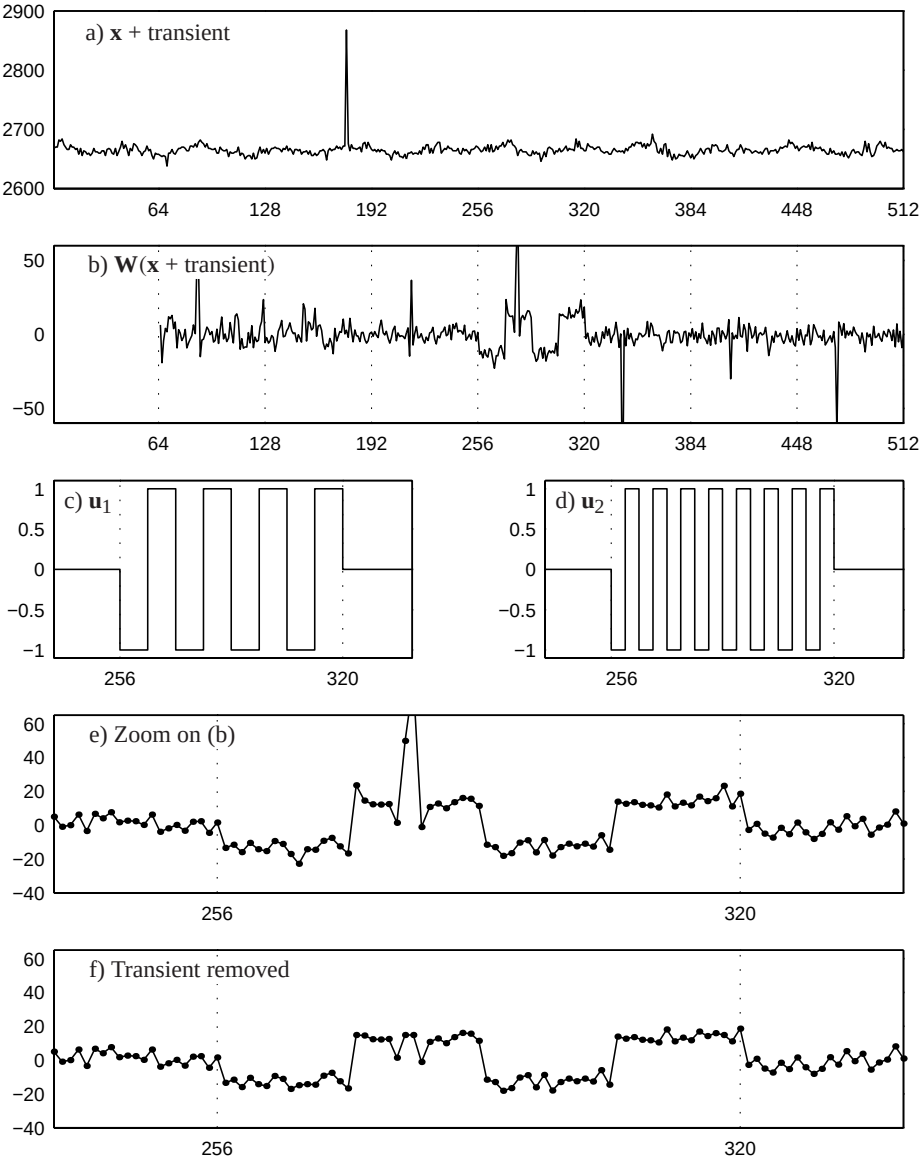


Figure 5.3: (a) the received signal with a transient added, (b) the transform of the signal in (a), (c) and (d) the two specially designed test signals, (e) a magnification of the interesting part of (b), (f) the result of using the first denoising method.

Table 5.6: Results of applying the solution method of Section 4.7.4.

Description	Fig	SNR (dB)	$\lvert\Delta G\rvert$ (%)	Inner product of $\mathbf{y}$ and $\mathbf{u}_0$	$\mathbf{u}_1$	$\mathbf{u}_2$	
No transient	5.2(d)	-3.7	1.1	851	4	59	$p = 0.93$
Transient at 177	5.3(a)	-8.0	13	947	-109	172	$p = 0.77$
On [257; 288]	5.3(e)		23	516	-135	134	$p_1^0 = 0.66$
On [289; 320]	5.3(e)		2.3	431	26	38	$p_1^1 = 0.87$
On [257; 272]	5.3(e)		2.7	216	-13	24	$p_2^0 = 0.85$
On [273; 288]	5.3(e)		43	301	-122	110	$p_2^1 = 0.56$
First method	5.3(f)	-0.72	2.9	817	4	59	$p = 0.93$

the accuracy of the estimated G is less than in the first row. This is also evidenced by p which has dropped to 0.77. The true error on G is 13%.

Note that the p value can be used as an indicator of the accuracy, but not as a quantification of the accuracy. Actually, it is indeed possible to have a signal full of transients and still having p close to 1. It is highly unlikely, but it is possible.

The design of $\mathbf{u}_1$ and $\mathbf{u}_2$ was not just an easy way of getting orthogonal signals. They were also designed in accordance with the design rule laid out in Section 4.7.4 which allows for p values on fractions of the transmission signal. When this method is applied to the present signals it yields $p_1^0 = 0.66$ on the first half and $p_1^1 = 0.87$ on the second half (the inner products in the third through sixth rows of Table 5.6 are all on the specified intervals). This indicates that the disturbance has occurred in the first half, and that the estimate based on the second half should be reasonably good (which indeed it is). Applying the method once more, this time to the first two quarters, locates the noise occurrence in the second quarter, and since $p_2^0 = 0.85$ the G estimated based on only the first quarter should also be reasonably good.

It has now been established that there is some kind of disturbance on the second quarter of the signals. Assuming that this disturbance is at most a few transients, and not a total corruption of the second quarter, there are several methods for removing the noise. One possible solution is to estimate G and then look for samples that deviates significantly from $G\mathbf{u}_0$. These samples are then reset to the value they should have had according to $G\mathbf{u}_0$. An abnormal value $y[n]$ can be defined as being larger than $C \cdot Gu_0[n]$ for some constant C. Applying this method for $C = 2$ to the signal in Fig. 5.3(e) yields the signal shown in Fig. 5.3(f). The new estimate of G, see the last row of Table 5.6, is improved significantly compared to the original estimate. However, it is not better than the estimates based on those fractions of the signal which was undisturbed by the transient. Note that it is not an error that $p = 0.93$ in the first and last row even though the $\mathbf{u}_0$ inner products yields 851 and 817. This anomaly is caused by 'unfortunate' rounding.

An alternative to denoising the transformed signal is to remove transients directly from the received signal. As before abnormal samples are found in the transformed signals, but instead of resetting this sample the transient from which it originates is found in the received signal. This is easy since the location of the transient in the element is

approximately equal to the location of the transient in the received signal (when the two signals are regarded as having the same time range). It it therefore sufficient to search a few samples in the received signal to find the transient. After resetting this samples the WPT is applied again. If the estimated G is still not satisfactory the search for another transient begins.

It is possible to avoid the repeated transformation by adding appropriately scaled low and high pass filter taps in the right fashion to the signal. This approach will not be discussed in this thesis.

The result of applying this second method is shown in Table 5.7 and Fig. 5.4. Three

Table 5.7: Results of applying the second transient removal procedure.

Description	Fig	SNR (dB)	$\lvert\Delta G\rvert$ (%)	Inner product of $\mathbf{y}$ and $\mathbf{u}_0$	$\mathbf{u}_1$	$\mathbf{u}_2$	$\mathbf{u}_3$	p
Transients	5.4(a)	−12	34	1125	−180	167	54	0.73
Step 1		−12	25	1052	−108	101	123	0.75
Step 2	5.4(d)	−10	13	952	10	−17	1	0.97
Step 3		−9.1	4.1	876	−26	20	32	0.92
Step 4	5.4(e)	−3.5	0.1	842	8	57	0	0.93
Step 5		−3.5	0.7	835	14	64	−7	0.91
Step 6		−3.5	2.0	825	4	53	−10	0.93
Step 7		−3.5	2.9	817	11	61	−2	0.92

more transients have been added to the signal which is shown in Fig. 5.4(a). The result of the WPT is shown in (b). The estimate of G is not very good, as the relatively low p witnesses. The p is given by the obvious formula

$$p = \frac{\langle \mathbf{y}, \mathbf{u}_0 \rangle}{\sum_{n=0}^{3} | \langle \mathbf{y}, \mathbf{u}_n \rangle |} .$$

After removing the most predominant transient the accuracy does improve, and removing one more transient reduces the inaccuracy to 13%. Notice how the p in this case is close to 1 which could lead one to believe that a very good estimate has been obtained, though it is still more than 10% wrong. As more test signals are used for estimating the accuracy the risk of such misleading information decreases, but even though an extra test signal has been included here, there are still significant risk of getting a high p value for rather inaccurate estimates of G. Note that the opposite is also possible, i.e. having a low p value even though the estimate is good.

The first seven steps of the iterative process of removing transients are shown in Table 5.7. In this case there are only four transients, and consequently, after four iterations the estimate does not improve.

The primary advantage of this method is that the transients in each element of the transformed signal are removed. This can be seen in Fig. 5.4(d) and (e), where first one transient and then the remaining transients disappear in the element to the left of the chosen element.

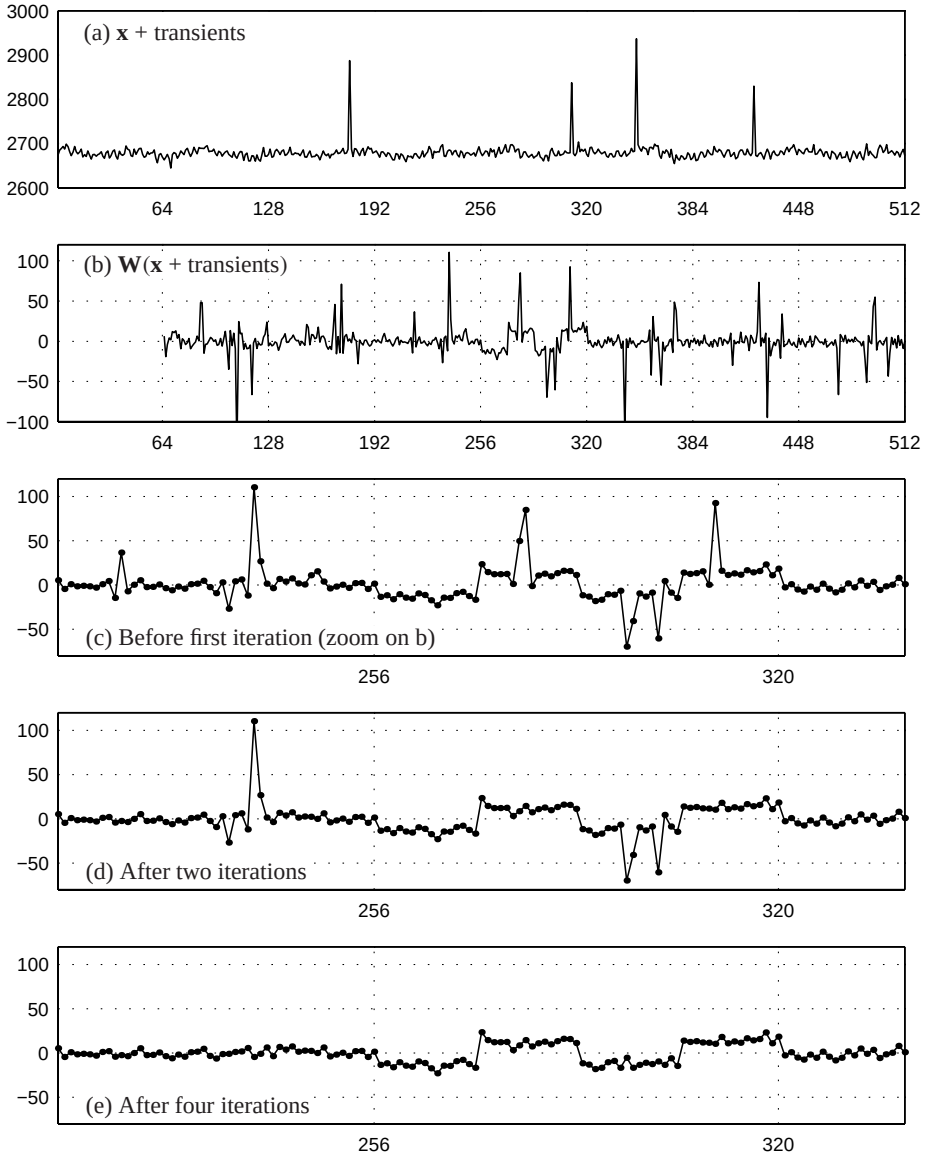


Figure 5.4: (a) the received signal with four transients added, (b) the transform of the signal in (a), (c) a magnification of the interesting part of (b), (d) and (e) the result of applying the second denoising method after two and four iterations.

5.3 Second Test Setup

The first test setup employed the wavelet modulation for measuring the channel gain. In this setup the Rudin-Shapiro spread spectrum modulation is used. The primary purpose is to demonstrate and evaluate the validation methods presented in the previous chapter.

The test setup used is actually designed for recognition of objects (see Section 7.6.1), and the measurement of channel gain is in that context merely a tool for acquiring information about the object to be recognized. The setup has three emitter and three receivers, all facing in the same direction, and each is connected separately to the PC. Thus a total of nine CGMs are generated simultaneously. The setup has been constructed in engineering and financial collaboration with LEGO Engineering, Denmark, and it is shown in Fig. 5.5.

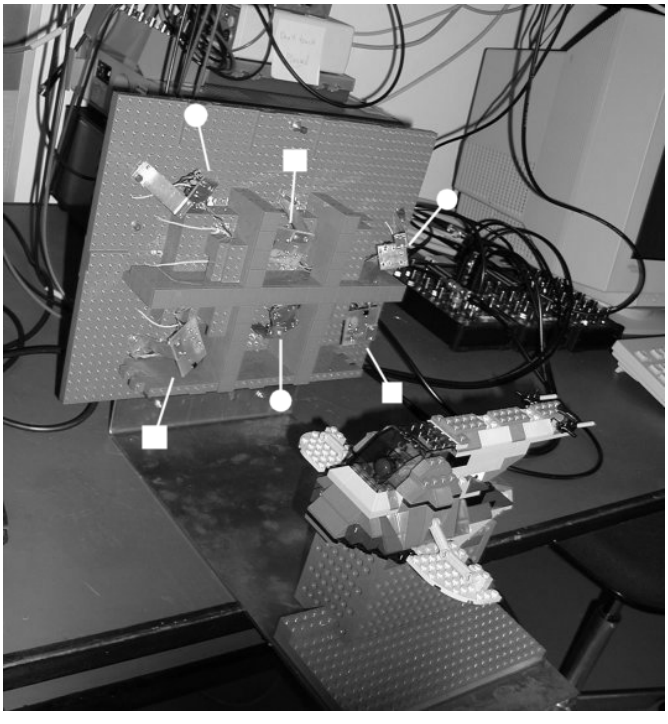


Figure 5.5: The second test setup was constructed in collaboration with LEGO Engineering. There are three emitters and three receivers facing forward towards the LEGO model. The setup is connected to the computer on the right. Three circles show the emitters, and the three squares show the receivers.

The emitter and receivers are infrared, like in the first test setup. The sampling frequency is the same order of magnitude as in the first test setup, but the comparatively

short length 16 RS sequence that is used means that the frequency of signals transmitted is almost 20 times higher, namely 165 Hz. While the setup is capable of generating nine CGM, only one is analyzed here. The separation of the signals is described in the previous chapter and is a straightforward procedure. It will therefore not be discussed any further.

5.3.1 Validation of Measurements

This test setup is used mainly for evaluating the validation methods presented in Section 4.9. For that purpose three test signals have been recorded. Note that test signal here refer to signals recorded with the test setup, and not designed signals described in the previous section and previous chapter. The signals are all from the same receiver (in Fig. 5.5 it is the one on the bottom right). The three emitters (from right to left in Fig. 5.5) are transmitting signals on RS channel 0, 1, and 2, respectively. There are 16 samples in each block, so there are 3 signal channels and 13 noise channels. All channels of each of the three received, transformed signals are shown in Fig. 5.6 and 5.7. The signals have been generated in the following way:

Test signal 1: A hand has been moved slowly into the space in front of the emitters and receivers, and then, at 3 seconds, moved out. Then it has quickly been moved back in and out twice, and finally back in. The hand has been 'inserted' from the right and thus the contribution from the left-most emitters has a smaller amplitude.

Test signal 2: The hand has now at the beginning been moved into the space in front of the emitters and receivers and out again. Then at 4 seconds it has been moved back in. Meanwhile the receiver circuit has been subjected to an electrical disturbance (by quickly touching on of the pins on the photodiode with a screwdriver).

Test signal 3: Again a hand has been moved in front of the emitters and receivers. This time somewhat closer than in the previous two signals. For 2.5 seconds the screwdriver has been touching the photodiode pin.

It is immediately clear from these test signals that there is a potential for detecting the noise occurrences in the second and third test signals. While the noise channels in the first test signals are pure white noise (this is evident from the histograms in Fig. 5.8) the noise channels in the two other test signals show clear evidence of inflicted noise. The question is now whether the two validation methods are capable of classifying each sample in the test signals correctly. It is possible to apply the validation methods to all 16 channels jointly, but the theory to support this approach has been left as future work. In this thesis the validation methods are applied to each channel separately. Actually, the validation is applied only to y_1, the second of the three signal channels, as the procedure is exactly the same in all cases.

The two validation methods use a threshold on the ratio between signal and noise to classify the CGM. When adapted to the test signals at hand the first validation method is

$$\mathcal{T}(\alpha): \quad \Theta(\mathbf{y}) \equiv \frac{y_1^2}{\sum_{k=3}^{15} y_k^2} > \alpha$$

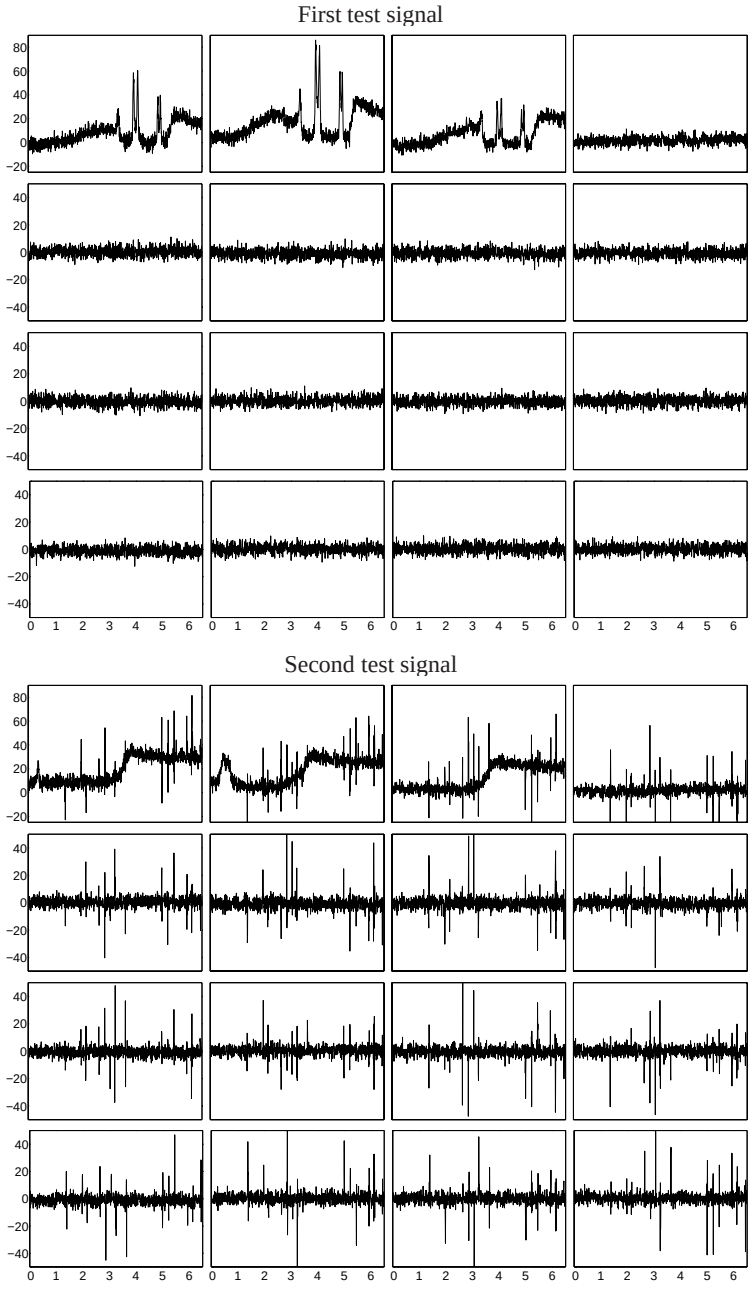


Figure 5.6: All 16 channels y_0 through y_{15} of the first and second test signals the first test setup. Note that the unit in the horizontal axis is seconds.

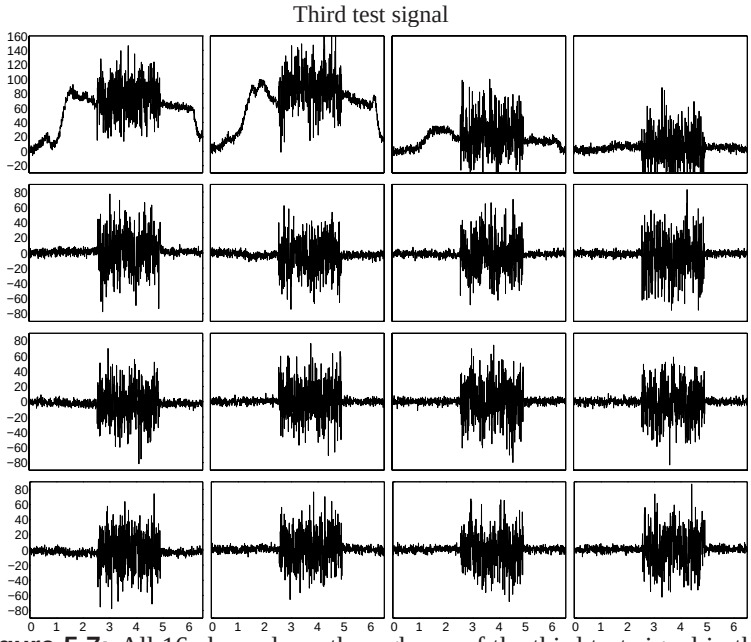


Figure 5.7: All 16 channels y_0 through y_{15} of the third test signal in the first test setup. Note that the unit in the horizontal axis is seconds.

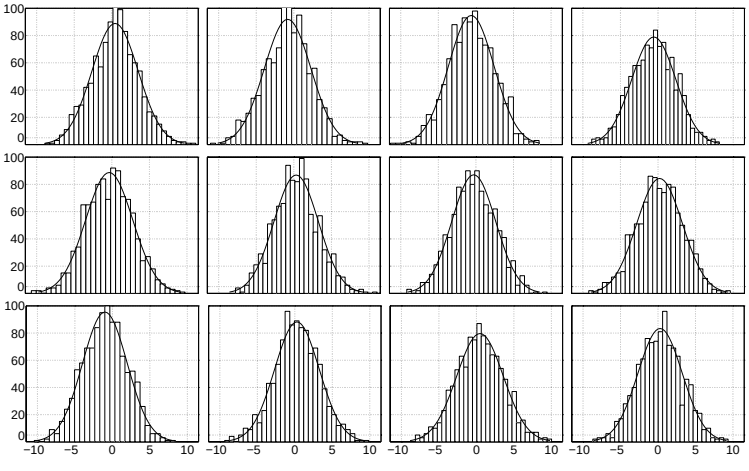


Figure 5.8: Histograms of y_4 through y_{15} in the first test signal of the first setup with a fitted normal density functions superimposed.

while the second method is

$$\tilde{\mathcal{T}}(\alpha): \quad \tilde{\Theta}(\mathbf{y}) \equiv \frac{y_1^2}{\sum_{n=3}^{15} y_n^2 + \beta|y_1|^3} > \alpha \ .$$

Both methods have been applied to the three test signals.

5.3.2 Applying the First Validation Method

The first thing to do is determine worst-case SNR, i.e. the weakest detectable signal compared to the expected random-noise level. The weakest signal is chosen to be $y_{\text{low}} = 15$. In the present setup this depends on the maximum distance at which a given object should be detectable. The first half of the y_1 channel in the first test signal is generated by moving an object from far away into the maximum detection distance and a little further in, thus showing the weakest detectable signal to be approximately 15. Since the variance σ^2 is approximately 9.1, $R_{\max} = y_{\text{low}}/\sigma = 4.8$. Using the curve the lower-most plot in Fig. 4.11 on page 79 this yields approximately $\alpha = 0.5$ for which $P_{\text{FP}} = P_{\text{FN}} \approx 0.02$.

The first test signal subjected to the first validation methods is shown in Fig. 5.9. The lowermost graph shows the validation function $\Theta(\mathbf{y})$ and the α is plotted as a dashed line in the same axis. The validation behaves as expected. For the 'no signal' part in the beginning $\mathcal{T}(\alpha)$ is false for almost all samples (approximately 1 out of 50 is expected to be accepted as a useful GM). When the signal level approaches y_{low}, which equals 15 and is shown with a dashed line, the validation shifts in favor increasingly more useful measurements. Again the fraction of measurements validated incorrectly is $1/50$ for signal level close to y_{low}. Note that the point in time (here measured in samples) at which more measurements are considered useful than not useful is the same as where the average level of the signal is $y_{\text{low}}/2$. This happens around sample number 250. This corresponds with the notion that if the signal itself was used for validation (as explained in Section 4.9) and P_{FP} should equal P_{FN} for the weakest detectable signal the threshold should be half the weakest detectable signal level. This level is 48 in the example in Section 4.9.

When the signal level then raises above y_{low} the P_{FN} decreases (but P_{FP} is still the same), and the last 200 samples of the signal shows that every single measurement is considered useful. Here P_{FN} equals approximately 10^{-4} (for signal level 27). Note also that the transient-like measurements are (correctly) considered useful by $\mathcal{T}(\alpha)$.

Now, applying the validation to the two other test signal and using the same parameters the result is less gratifying, see Fig. 5.10. The second test signal seems to be as expected, at least for the non-transient samples. Zooming in on the transients (not shown) reveals that a majority, but not all, of the transients have indeed be classified as useless. The third test signal shows this only too clearly. When not just a few, but 400 consecutive samples are transient noise, the incorrect validation becomes evident as the majority of these samples are classified as useful.

It is worth noting that the validation does not fail because there is no difference significant difference between $\Theta(\mathbf{y})$ for the useful and useless parts of the third test signal.

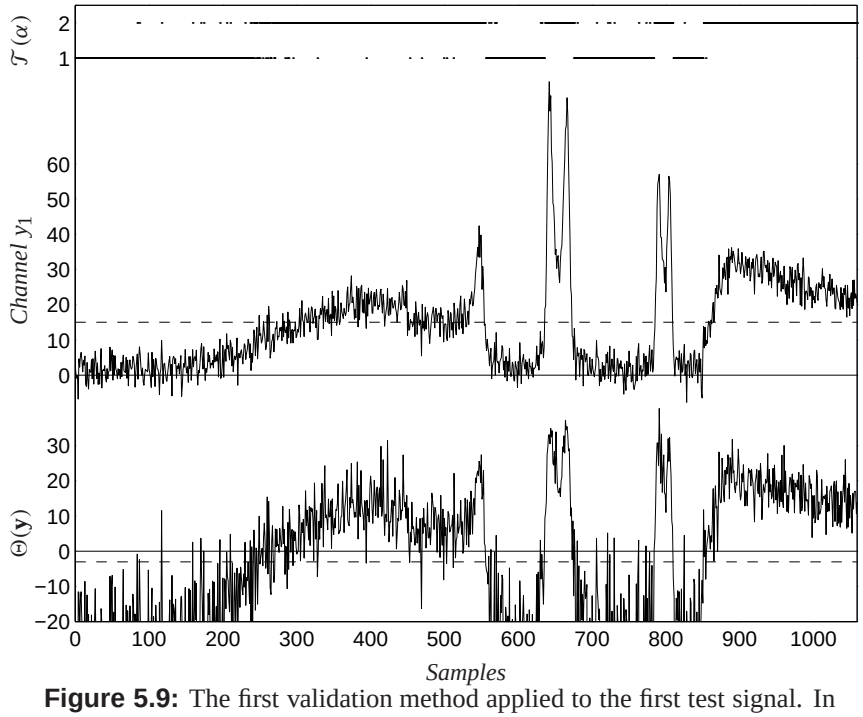


Figure 5.9: The first validation method applied to the first test signal. In the middle is a graph of the y_1 channel with a dashed line showing $y_{\text{low}} = 15$. The lowermost is a graph of the corresponding $\Theta(\mathbf{y})$ (in dB). The two lines on top show for each sample 1) when the test $\mathcal{T}(\alpha)$ is false and 2) true. Here $\alpha = 0.5$ (which is -3.0 in the dB scale of the above graph and marked by the dashed line) and $P_{\text{FP}} = P_{\text{FN}} \approx 0.02$.

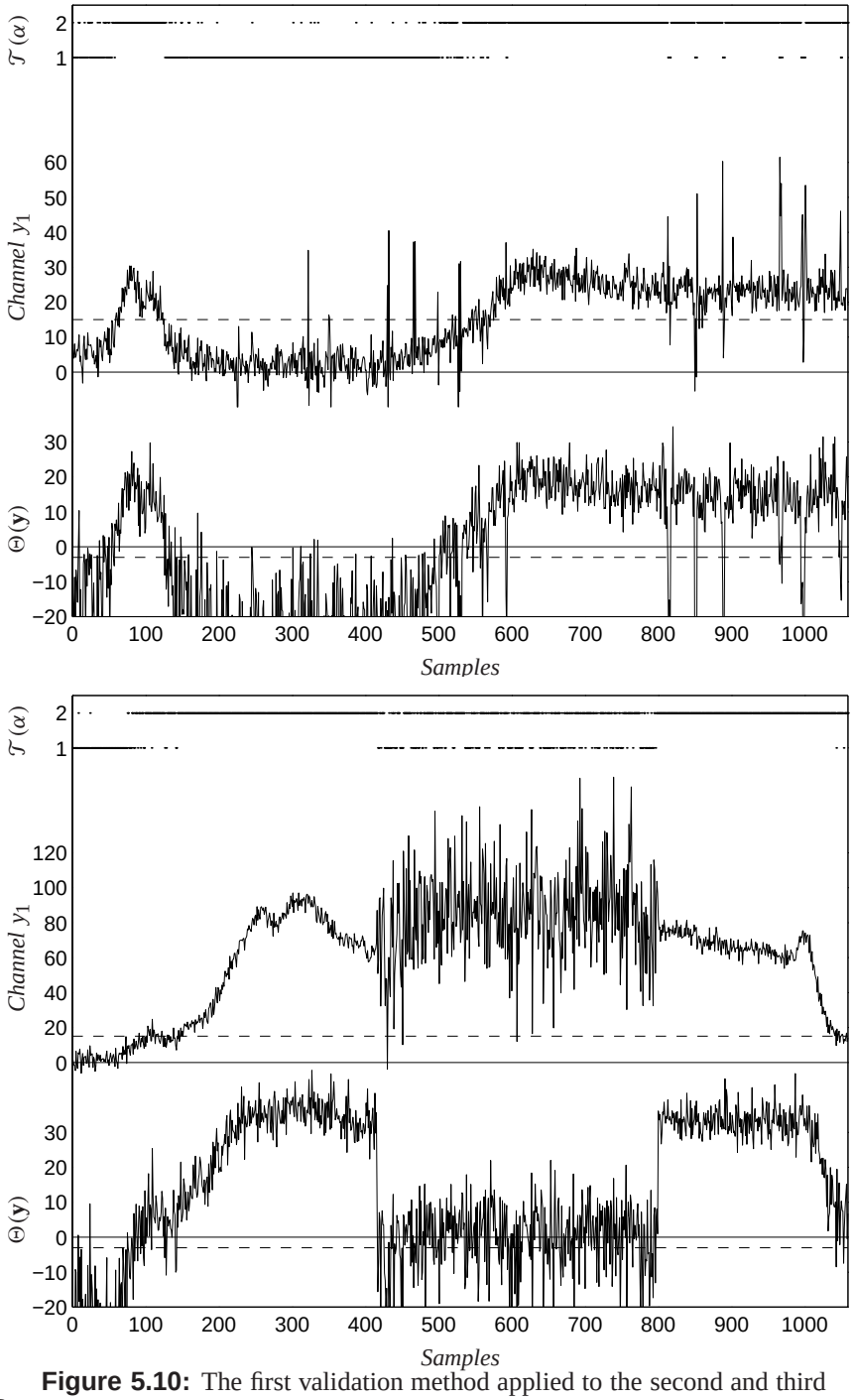


Figure 5.10: The first validation method applied to the second and third test signals.

It fails because the threshold is wrong. However, the threshold has been determined such that it complies with our notion of proper behavior in a random noise scenario, and accordingly it works fine for the signal without transients. This problem can be solved easily by increasing the threshold to around 20, which seems to separate nicely the useful and useless parts in the third test signal. However, the result of simply increasing the threshold (which applies to all three test signals) is that virtually all the samples in the first test signal are classified as useless. Evidently, another validation method is needed to handle this problem.

5.3.3 Applying the Second Validation Method

This is why the second validation method is relevant. The introduction of this second methods is solely an attempt to handle the transient in a proper manner while at the same time responding to random noise exactly as the previous method. In this method the S test parameters is included and β has to be determined, too. The S parameter is mainly to ensure that in the case where the signal is too weak for any reliable detection the validation function is not used (there is no reason to use a function than depends on a stochastic process if the measurement is indeed too small). For this test $S = 7.5$ which is half the weakest detectable signal level.

The next step is to determine α (see Section 4.9.6). Using the $P_{\text{FP}} = P_{\text{FN}}$ curve in Fig. 4.15 yields a value a little smaller than for the first validation methods, namely $\alpha = 0.6$. The β is determined according to the guidelines given in Section 4.9.6 with a maximum value of y set to approximately 100. This gives $\beta = 0.017$. To ensure a not too large value $\beta = 0.012$ is used. The result is shown in Fig. 5.11 and 5.12. It is clear that the results in the first two test signals are approximately the same as in the first validation method (though almost all the transients in the second test signal have been classified as useless by the second validation method), but the third test signal is now in general classified correctly. More of the transients can be 'caught' by increasing β, but as described previously this lowers the maximal acceptable value of y_1. In this case experiments have shown that increasing β to 0.016 will produce a very good result. However, if this value is chosen, new y_1 signals must not exceed 100, or they will be classified as useless.

It is interesting to note that while the chosen values yields a $P_{\text{FP}} = P_{\text{FN}} \approx 0.02$, choosing $\alpha \approx 2$, which corresponds to an SNR of approximately 10 and thus a weakest detectable signal of $10 \cdot 3.1 = 31$, would actually be optimal in terms of balanced probabilities; for the given number of noise channels and for given β and noise variance it is simply not possible to achieve a better balanced error rate with the second validation method.

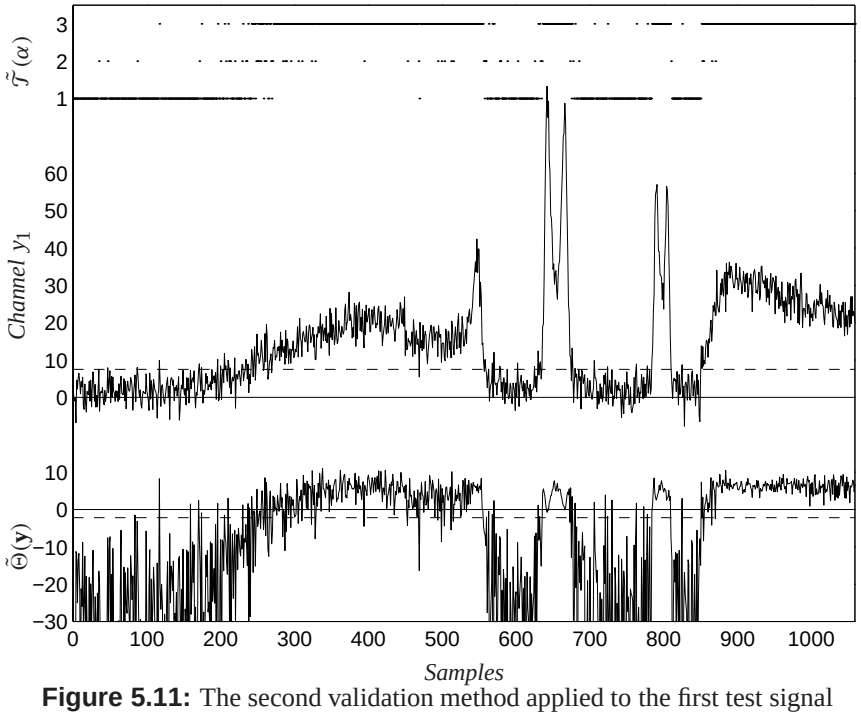


Figure 5.11: The second validation method applied to the first test signal in the second setup. In the middle is a graph of the y_1 channel with the S value plotted as a dashed line. The lowermost is a graph of the corresponding validation numbers $\tilde{\Theta}$ (in dB). The three lines on top show for each sample 1) when $y_1 < S$, 2) when the validation number is less than $\alpha = 0.6$ (which evaluates to -2.2 on the dB scale), and 3) the remaining cases which are the useful measurements.

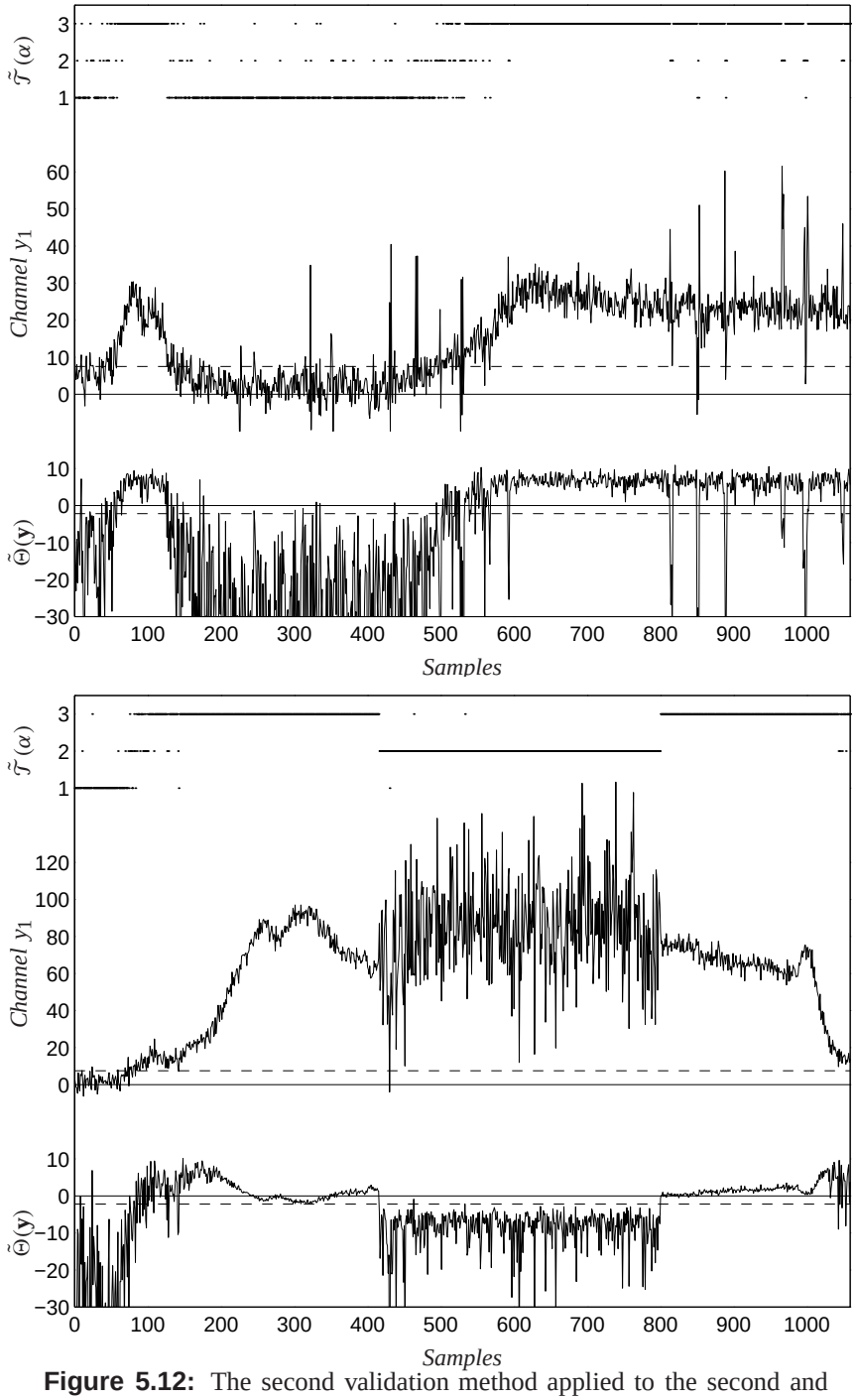


Figure 5.12: The second validation method applied to the second and third test signals in the second setup.

5.4 Third Test Setup

The first two test setups relied on a reflecting object for directing the light from the emitter onto the receiver. In this the third setup the emitter is facing towards the receiver and the light travels directly from emitter to receiver and the distance is approximately 3 meters. While the descriptions of the first two test setups did not include the hardware, the driver and amplifier circuits are presented for this setup. This includes a brief discussion of the internally generated noise, see Section 5.4.6.

The primary purpose of this setup is still the signal processing, though. Especially, this setup is used for evaluating the polynomial removal procedure introduced in Section 4.8. As this method applies to the spread spectrum type modulation, the Rudin-Shapiro transform is used for coding the transmission signal. It is interesting to examine the result of polynomial denoising in the typical noise cases, i.e. white noise, and frequency and time-localized noise. Therefore, a set of signals having such noise types are recorded and subjected to the polynomial removal procedure. The result is presented and discussed in Section 5.4.2.

First, a brief description of the polynomial removal procedure is given. The theory was presented in the previous chapter, and the following description is therefore simply the application of the procedure to a specific signal.

Note that as in the previous section the term ‘test signal’ refers to the signals recorded with the test setup.

5.4.1 Polynomial Removal Procedure

A signal containing a high amplitude low frequency disturbance has been recorded. The disturbance is the artificial lighting in the laboratory, and is approximately one magnitude more powerful than the transmitted signal. This recording is a small part of the signals presented in the next subsection (the 260th transmitted length 64 RS sequences in the second test signal), and is shown in the top graph of Fig. 5.13. It is decided that eight third degree polynomials should be fitted to the signal, and the result is the dashed line in the same graph. The difference is the thick line. It is clear that a major part of the energy has been removed from the signal by this procedure.

In the second graph in the figure is shown the result of transforming (with the RST) the received signal and the denoised signal. The transform is energy preserving and consequently the oscillations in the transformed non-denoised signal have a much higher amplitude. As the transformed signal is ideally all vanishing except for the first sample, it is obvious that the SNR is quite small without denoising. Removing the low-degree polynomial content makes a big difference as the energy is significantly reduced in the 63 noise channels. At the same time the first sample (which indicates the energy in the transmitted signal) has changed from about -60 to $+30$. A noiseless transmission yields a positive number, and -60 therefore indicates a major noise contribution while 30 is plausible as an estimated channel gain.

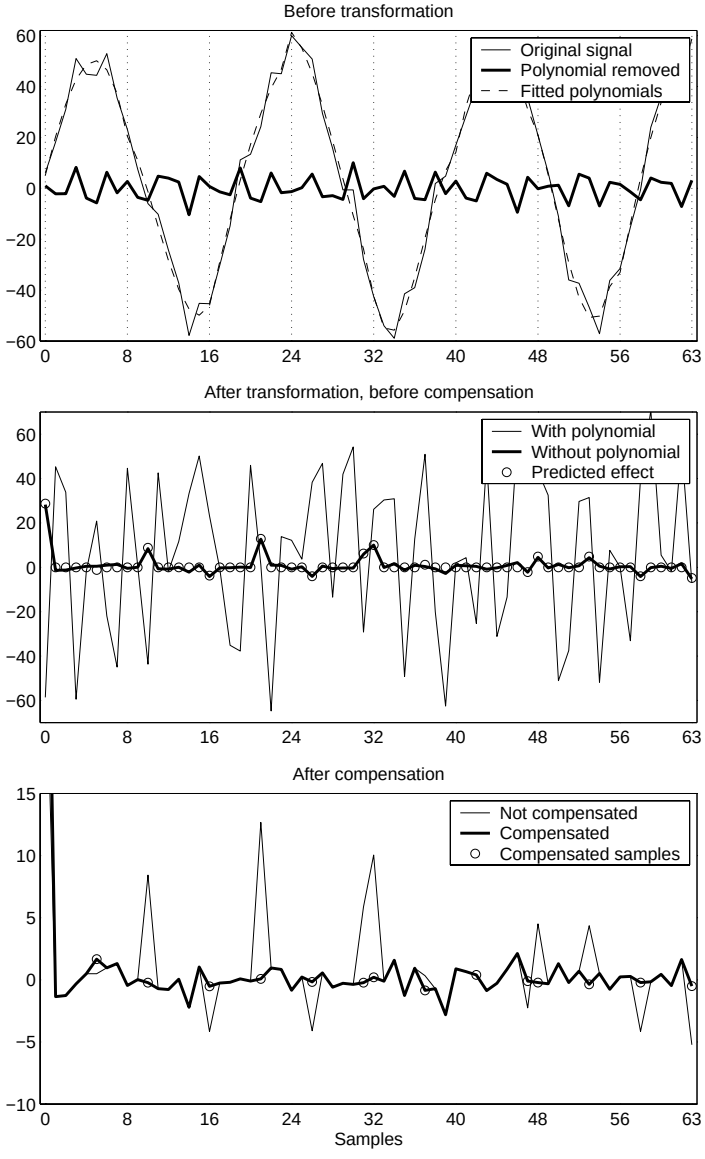


Figure 5.13: The original signal is the one received by the photodiode. Eight third degree polynomials are fitted to the signal and subtracted. Then below the signal is transformed (for comparison the signal with the polynomial still there are also shown after transformation). The effect described in Section 4.8.2 are also shown in this second plot. In the third plot the compensation, also described in the previous chapter, has been applied.

As described in Section 4.8.2 the polynomial removal affects the RS sequence in a predictable way. In the second graph in Fig. 5.13 the circles show the predicted effect, i.e. the shape a noiseless RS sequence subjected to the same polynomial denoising. This signal is normed to have the same energy as the denoised signal. The match is rather good, and thus the compensation, see Section 4.8.2, applied to the denoised signal by means of this predicted signal yields another signal which has very little energy in the 63 noise channels. This is shown in the third graph in Fig. 5.13.

The result of denoising the entire second test signal in this fashion is shown in Fig. 5.17. The highly oscillating graph in the top plot shows the received signal while the more 'calm' graph in the bottom plot shows the signal after denoising. To see the effect of compensation (this is not immediately evident in Fig. 5.17) the amplitude of the two of the most affected channels, that is number 10 and 21, is shown in Fig. 5.14 for a part of the second test signal.

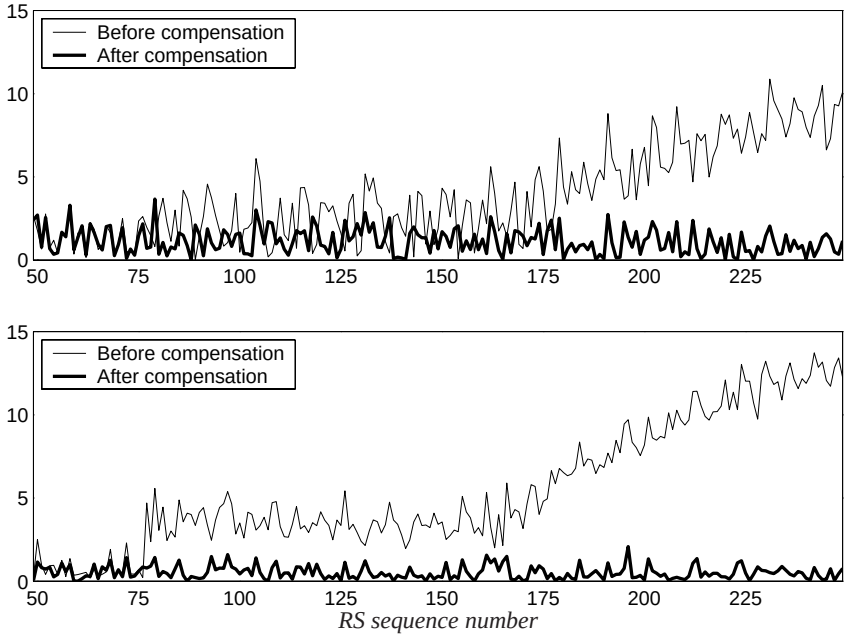


Figure 5.14: The energy in the 10th (top) and 21st (bottom) channels in the second test signal after polynomial removal, and without and with compensation.

5.4.2 Generating Noise for Test Signals

To be able to the compare the effect of polynomial denoising when different types of noise occurrences are present a set of six test signal has been recorded. The intensity of

the transmitted signal has been varied approximately equally in all six recordings. The graph in Fig. 5.15 shows the intended variation. The strong signal corresponds to the

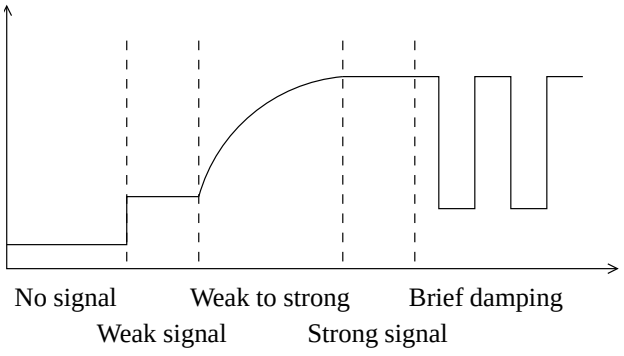


Figure 5.15: All the six test signals in the third test setup are recorded such that they roughly follow this pattern.

photodiode being fully exposed to the incoming light. The weak and weak to strong parts have been achieved by mounting the receiver circuit vertically and such that it can turn around its vertical axis. At the same time a small 30 mm high screen was mounted right next to the diode on the PCB. By rotating the circuit the screen covers a smaller or larger part of the photodiode, thereby weakening the signal. A complete covering results in the weak signal as some light finds its way around the screen by reflections. The brief damping in the end is achieve by moving a hand in between emitter and receiver. The very first part of each test signal shows the noise in the setup as the emitter here is turned off.

The test signals have been recorded in the presence of time-localized noise generated optically and electrically. The former with a remote control from B&O (this is a particularly powerful remote control) and the latter by touching one of the pins on the photodiode with a screwdriver. The two noise types plus the 'no noise' case has been recorded with and without the presence of artificial lighting. The lighting produces significant amounts of low frequency noise in the signals. A list of the test signals is given in Table 5.8 along with a figure numbers.

Table 5.8: List of noise types in the 6 test signals.

Number	Artificial light	Other disturbance	Figure
Test signal 1	No	None	5.16, 5.22
Test signal 2	Yes	None	5.17, 5.22
Test signal 3	No	B&O remote control	5.18, 5.23
Test signal 4	Yes	B&O remote control	5.19, 5.23
Test signal 5	No	Screwdriver on receiver circuit	5.20, 5.24
Test signal 6	Yes	Screwdriver on receiver circuit	5.21, 5.24

The figures 5.16 through 5.21 show the test signals after transformation along with the

validation according to the second validation method (see Section 4.9.6). These figures have the same structure as Fig. 5.11 showing one of the test signals in the previous section, i.e. for the second test setup. In all cases $\alpha = 1$, $S = 5$, and $\beta = 0.01$. Only S has been changes compared to the previous test setup. This is because the noise floor is lower in this setup.

5.4.3 Effect of Polynomial Denoising

The first test signal, see Fig. 5.16, does not contain any intentionally generated noise, and the noise is therefore (mainly) the shot and thermal noise from the photo detection part of the receiver circuit (see Section 5.4.6). It is therefore expected that the signal is classified as useful in all parts except in the no signal part in the beginning. This is almost the case as all but a few measurements have been classified as useful. Apparently, the abrupt changes in signal level has caused some of the measurements to have quite low rating in terms of the adapted SNR. The reason is a little more subtle, though. As a hand was moved in between the emitter and the receiver the overall light intensity on the receiver changed, too. The natural light in the laboratory is the main contributor to the current generated in the photodiode (even when the weather is cloudy), and this light comes from all directions as it is reflected by the walls, the floor, the equipment etc. When an object is moved in the close vicinity of the receiver the amount of natural light on the receiver changes a little, and sometimes sufficiently fast for some of the low frequency energy to 'escape' the DC removal in the receiver circuit. The consequence is a low frequency contribution in the received signal. This decreases the SNR and the affected samples are thus classified as useless. Incidentally, this phenomenon is, for obvious reasons, significantly reduced with the polynomial removal procedure. This is clearly seen in the polynomial denoised signal in the bottom plot in Fig. 5.16, where all the samples are now classified as useful (except for the first few samples).

Denoising the next test signal seems to be a bigger challenge, see Fig. 5.17. The artificial lighting of the laboratory causes a quite powerful 100 Hz harmonic to be present in the signal. This also causes an oscillation in the CGMs (see Fig. 5.17 and 5.22). Note that the oscillation frequency in the transformed signal is *not* necessarily equal to the frequency of the oscillations in the received signal. This oscillation in the transformed signal depends on the phase shift of the 100 Hz oscillation from received signal block to block. The sampling frequency is 1950 Hz so there are 19.5 sample per 100 Hz oscillation in the received signal. This means that there are $64/19.5 = 3.282$ oscillations per signal block. The fractional part of this number indicates the phase shift between consecutive signal block, and it takes $1/0.282 = 3.5$ blocks to do a complete 2π shift. Thus, every third and a half signal block is approximately equal in terms of the dominating harmonic, and the estimated CGM is for these signals approximately the same (wrong) value. The CGM therefore oscillates with a frequency of $3.5 \cdot 1950/64 = 106.6$ Hz. This observation should not be confused with the fact that $64/19.5 \approx 1/0.282$, i.e. that the reciprocal of the fractional part of $64/19.5$ is approximately equal $64/19.5$ itself. This is merely a

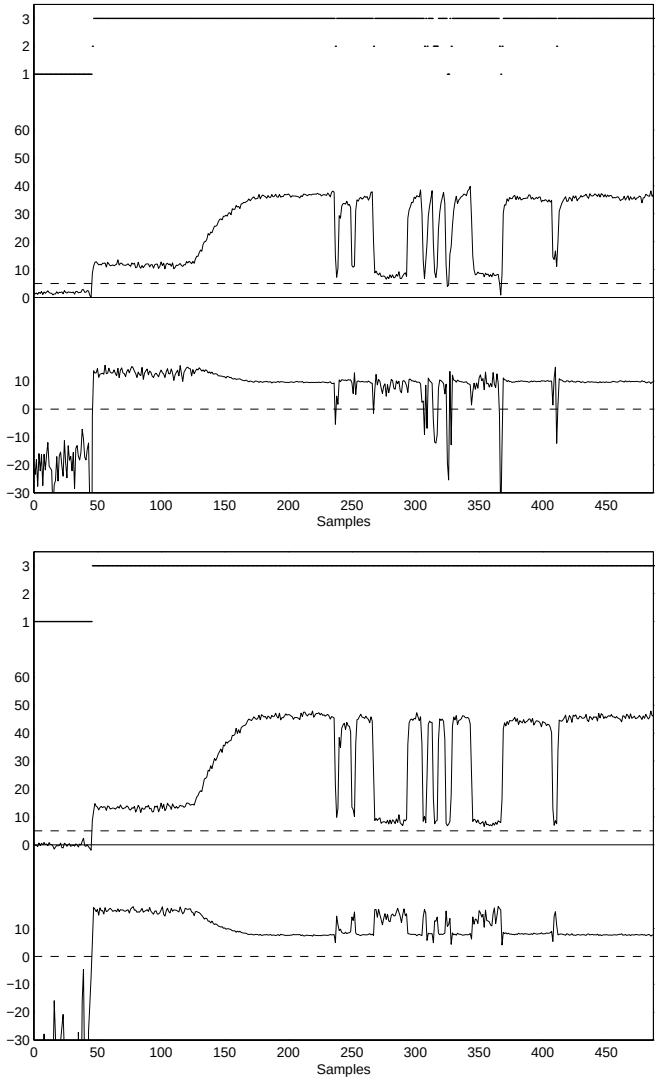


Figure 5.16: This and the following 5 figures shows the same as Fig. 5.11, but for the test signal of the third setup. This is the first test signal. The top plot shows the signal without polynomial content removed, the bottom plot shows the signal where eight times third degree polynomial content has been removed followed by compensation. Note that all test signals are shown after transformation.

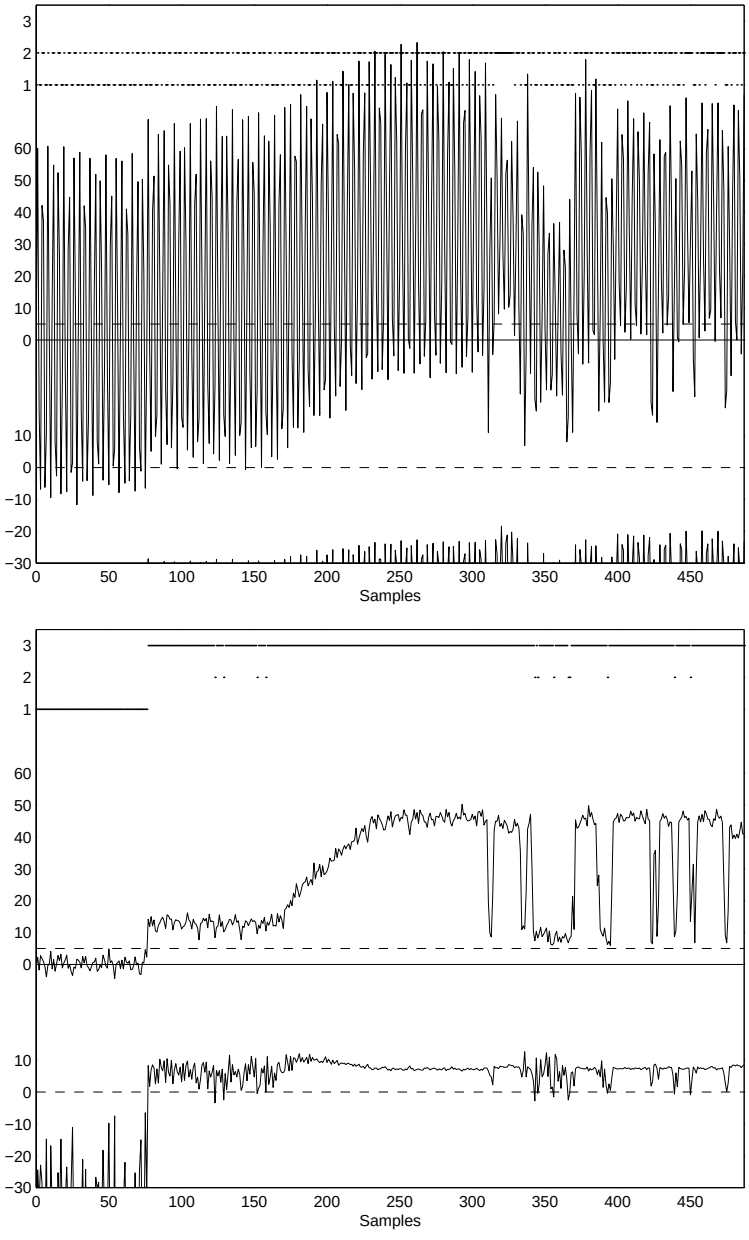


Figure 5.17: Test signal 2. With artificial lighting.

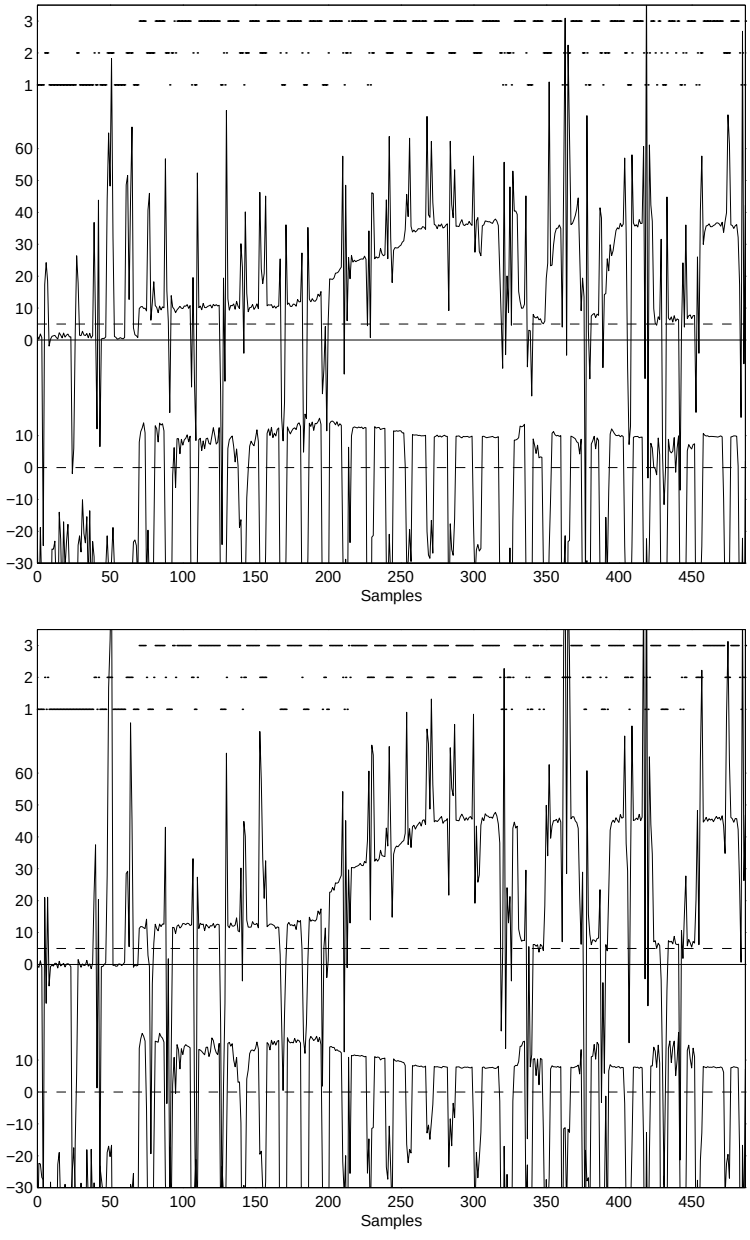


Figure 5.18: Test signal 3. B&O remote control noise.

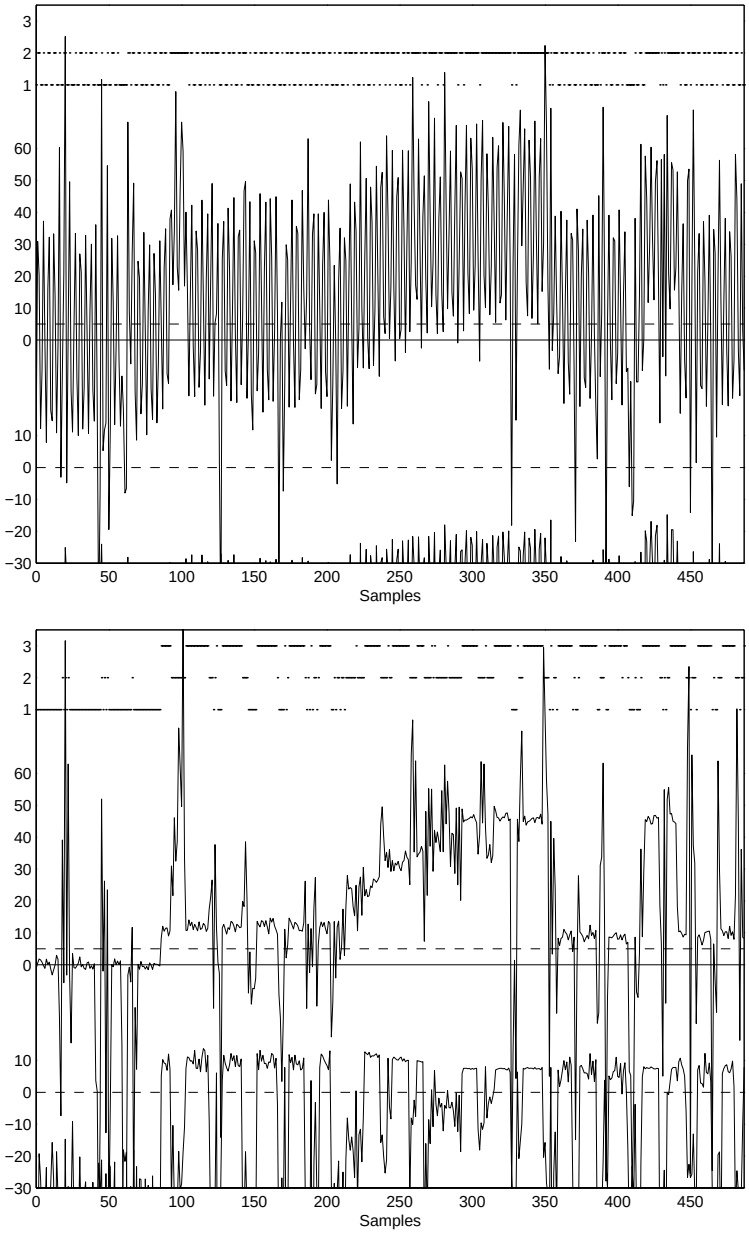


Figure 5.19: Test signal 4. B&O remote control noise and with artificial lighting.

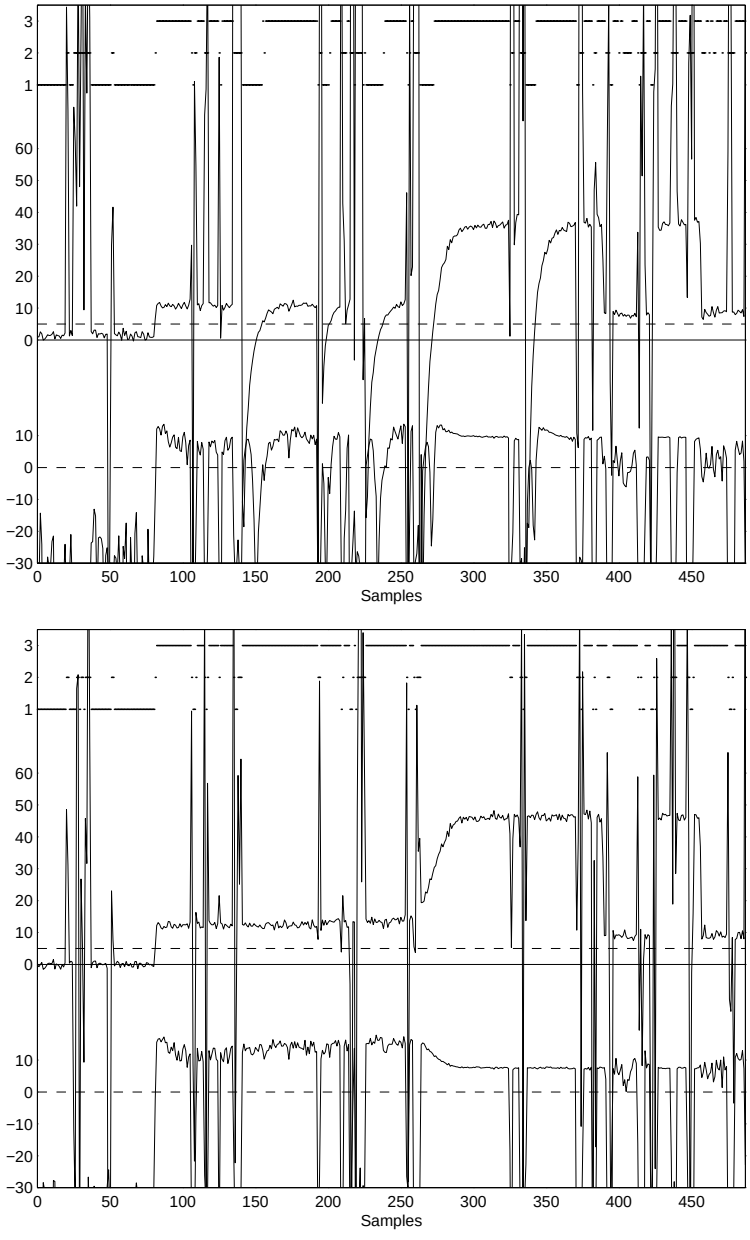


Figure 5.20: Test signal 5. Screwdriver on receiver circuit.

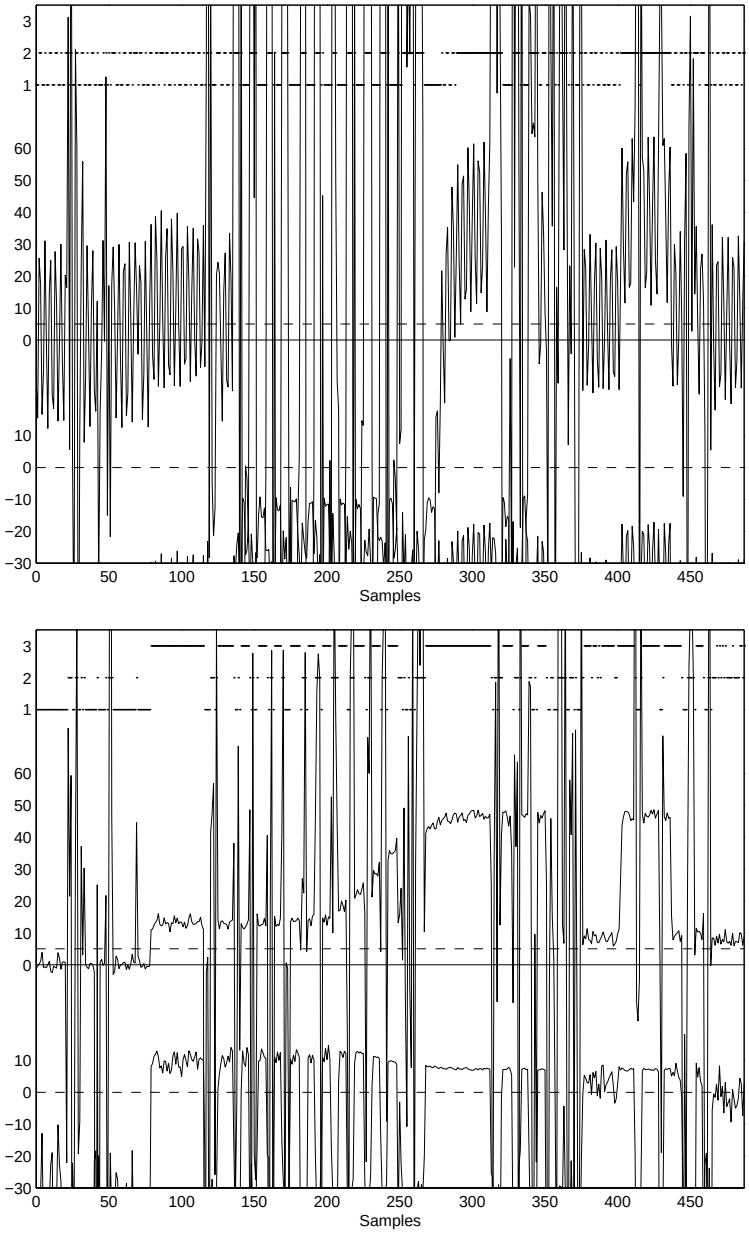


Figure 5.21: Test signal 6. Screwdriver on receiver circuit and with artificial lighting.

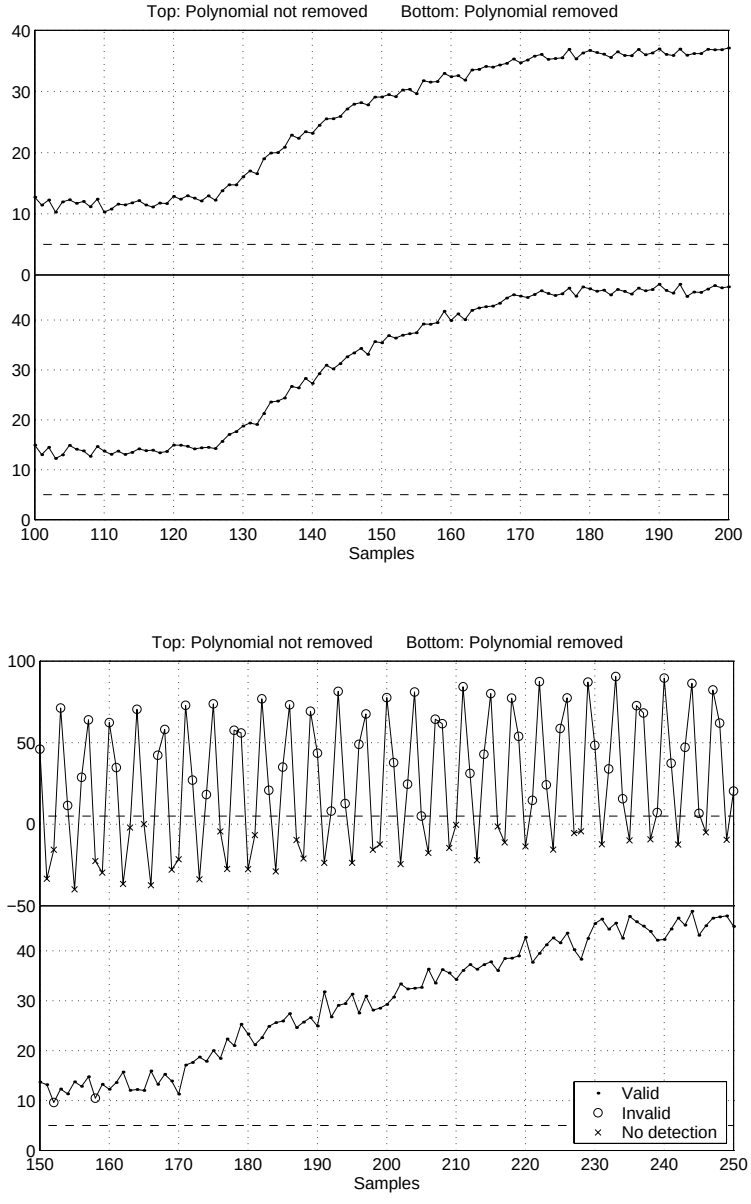


Figure 5.22: Zoom on the test signals 1 and 2 with the three lines of dots now applied directly onto the signal. Note that the $\tilde{\Theta}$ is not shown in this plot.

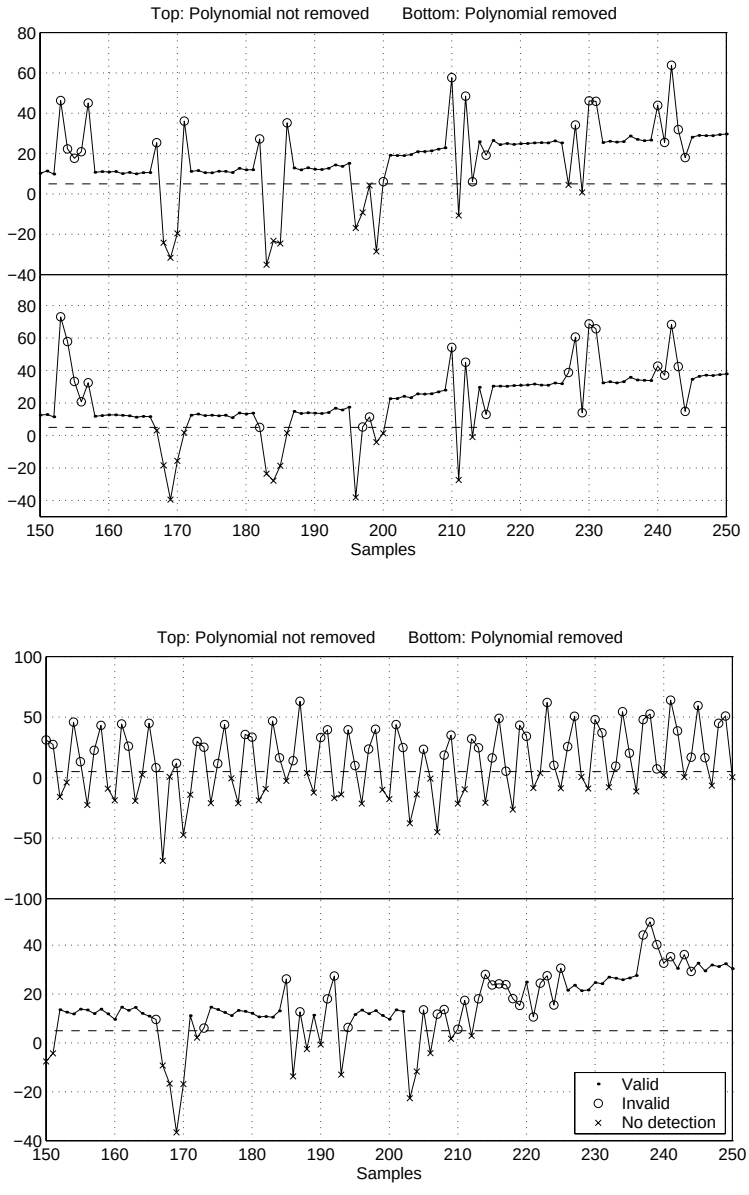


Figure 5.23: Zoom on test signals 3 and 4.

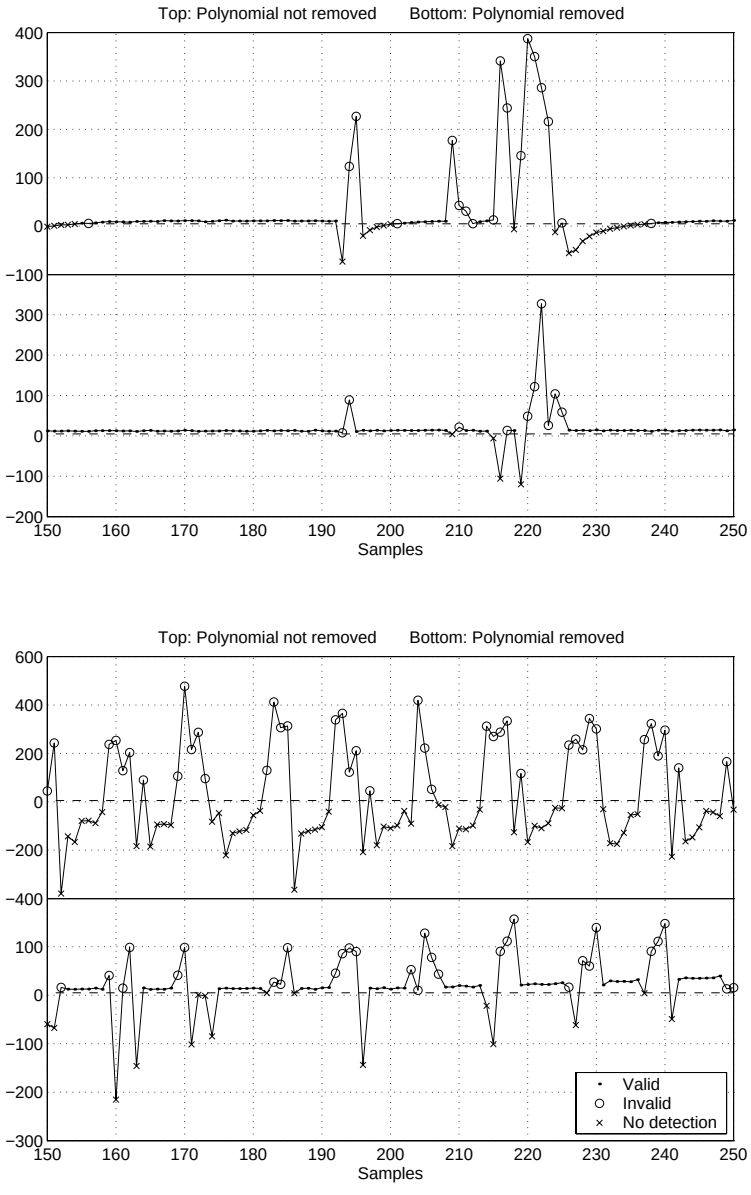


Figure 5.24: Zoom on test signals 5 and 6.

coincidence! This becomes evident when one does the same calculations with a sampling frequency of, say, 1600 Hz.

Though these observations are interesting they still leave the challenge of removing the 100 Hz from the received signal. The very same polynomial removal procedure as applied to the first test signals is used here. The result is shown in the bottom plot of Fig. 5.17 and 5.22. Evidently, the main noise energy has been removed without significantly disturbing the transmitted signal. The SNR is somewhat lower compared to the first test signal, and consequently a few samples are classified as useless. In general, the result of polynomial removal is quite good, both in terms of the CGMs and the ability of the method to validate samples correctly. It is still unclear, however, whether truly corrupted samples will be handled correctly, i.e. classified as useless, when high amplitude low frequency noise is present.

5.4.4 Polynomial Denoising when Transients are Present

To examine the behaviour of the polynomial denoising when transients are present in the signal the remaining four test signals contain two different types of transient energy. Two of the signal have low frequency energy and the two others do not. The first of these signals is the third test signal in Fig. 5.18. A number of transients have been generated using an infrared remote control close to the receiver. The remote control uses short bursts of modulated signals to transmit information, and the result is that a few consecutive samples in the transformed signal is disturbed (for each burst). The bursts are quite powerful and renders the received signal useless for determining the CGM. The second validation method were presented in Section 4.9.6 in the previous chapter, and in Section 5.3 in this chapter, and will not be discussed any further here. The focus in this section is on the consequences of polynomial denoising.

There does not seem to be much of a difference between the two signals in Fig. 5.18, and the validation rates the CGMs approximately equally. It is more interesting to examine the same transient noise in the case where low frequency noise is also presented. An example of this is shown in Fig. 5.19. The received signal is clearly useless for estimating channel gain, and every single sample is accordingly classified as useless (or too small). The polynomial denoising is capable of removing the main part of the low frequency energy and restore the non-transient parts of the signal. This is not surprising, though, as the denoising is applied to each RS sequence individually and each sample in Fig. 5.19 represents one sequence. That is, the samples in the figure is independent of each other and it is therefore possible to polynomial denoise a sequences (i.e. a sample) even though the one before and the one following it are both subjected to transients. It is not easy to determine by means of the figure how the transients have changes under the denoising, since it is hard to tell by the top plot which samples are indeed corrupted. The previous figure tells this story, however. Since the transform and the denoising are linear operations the transients and the low frequency noise can be regarded separately. Notice how the usable parts of the signal in the lowermost plot of Fig. 5.19 exhibits the same slightly

more white noise-like structure that was found in second test signal after denoising.

Another type of time-localized disturbance is shown in Fig. 5.20. The disturbance is not applied optically, but electrically. While optically generated noise only generate positive transients (as the extra light only causes more, not less energy) electrically generate transients can be positive as well as negative. This type of noise can also cause other effects such as shift of DC level and increased white noise energy. In contrast to the test signal in Fig. 5.18 there is in this case a significant difference between the signal before and after polynomial denoising. Touching the photodiode pin with a screwdriver has caused not only transients, but also a major change in the signal level. As the circuit stabilizes the signal level returns to normal and a charge curve shape (exponential) is clearly seen following some of the transients in the signal. Such a slowly change obviously qualifies a low frequency component, and it thus removed by the polynomial denoising. The result is that more of the samples are classified as useful.

The combined effect of transients, the 100 Hz harmonic, and charge curve shapes is found in the sixth test signal. Due to the linearity of the algorithm it is easy to predict the effect when polynomial denoising is applied. And indeed, apart from the samples corrupted by transients the signal exhibits the structure of weak and strong signal parts which it is supposed to do.

5.4.5 White Noise

The random noise found in any electromagnetic receiver is obviously also present in the test signals of the third setup. In the previous test setup the noise was analyzed in the individual channels. Instead of showing the same plots for this setup Fig. 5.25 shows the distribution of the noise samples from all the 63 noise channels. A total of 1.7 million samples have been recorded and the quality of the fit between the distribution of the samples and the solid curve shows that they are very close to being normally distributed. A small amount of outliers have been recorded, too. They are distributed fairly evenly throughout the signal and the origin of these noise samples is unknown.

Another interesting graph is the distribution of the denominator of (4.3) on page 51 in the presence of white noise. It has been stated in the presentation of the validation methods that this quantity is χ^2 distributed with $K + 1$ degrees of freedom. This can be visually verified in Fig. 5.26. The few outliers in the noise recording causes the estimated p.d.f. to be shifted slightly to the right of the histogram of the sums-of-squares.

5.4.6 Hardware in Setup 3

The emitter and receiver circuits used in the third test setup both have a fairly basic construction. Obviously, it has been attempted to achieve well-designed circuits with few components, but since they are meant to serve in various other setups there has been no attempt to adapt them specifically to the transmission conditions in this particular setup. The reason for introducing them in this thesis (which is definitely not about analog hardware) is on the one hand to show readers with some interest or knowledge in basic elec-

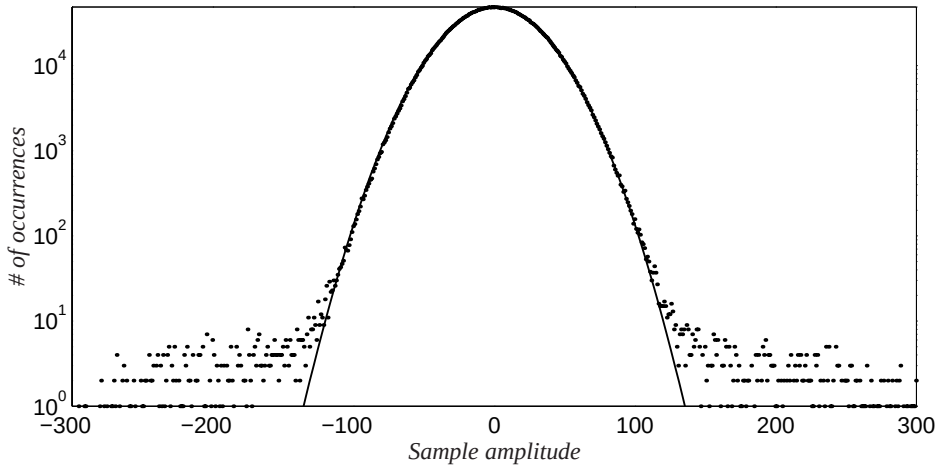


Figure 5.25: The dots shows the histogram (bin width is 1) of the 64 channels together for 15 minutes of noise (no transmitted signal and no artificial lighting) in the third test setup. The solid line shows the normal pdf with mean and variance estimated from the signal. The vertical axis is logarithmic.

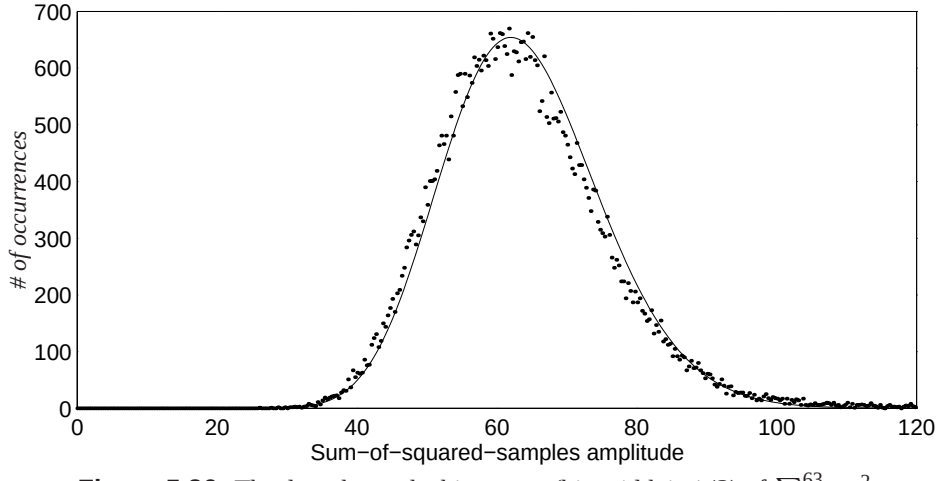


Figure 5.26: The dots shows the histogram (bin width is $1/3$) of $\sum_{n=0}^{63} y_n^2$ for the noise shown in Fig. 5.25 where each channel has been shifted and scaled to have zero mean and unit variance. The solid line shows the χ^2 distribution with 64 degrees of freedom scaled to correspond to bin width and the number of samples. Note that this plot is the same before and after a orthogonal (i.e. energy preserving) transformation of $\mathbf{y}$.

trical circuits exactly what has been used to record the signals analyzed above. And on the other hand to allow for a quantitatively analysis of the internal noise conditions in the receiver.

The emitter circuit is shown in Fig. 5.27. The circuit is designed to convert voltage input in the range 0 to 5 V to a current through the LED in the range 0 to 1 A. The LED is the near-infrared emitter SFH 487 P with peak wavelength at 880 nm which is just above the visible wave length of red light. This emitter has been chosen because of its size ($3 \times 4 \times 4$ mm) and directional characteristics (± 65 degrees half angle). While the radiant intensity at 100 mA, the maximal continuous operating current, is quite small the intensity with pulsed current, $\leq$ 100 μs at 1 A, is 15 times higher, namely 30 mW/sr. The large half angle ensures that the emitter does not have to be accurately faced towards the receiver. Deviations of up to 20 degrees are hardly noticeable since the directional characteristics is approximately a cosine.

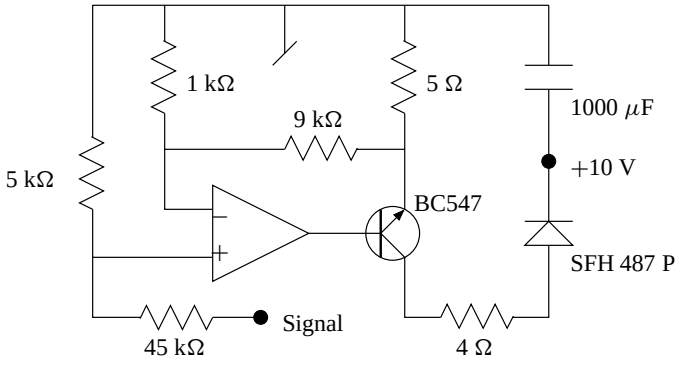


Figure 5.27: The emitter circuit for setup 3.

The receiver circuit is shown in Fig. 5.28. This circuit uses three coupled transistor to achieve a specified amplification. Alternatively, an integrated amplifier could have been used. This particular designed in found in an application note by Hyder [44]. The main part of the circuit is the amplifier including a few filters. The photo detection part of the circuit is the photodiode and the load resistor. The amplifier amplifies the voltage over R_{load} which is proportional to the current generated by the diode. The output at Signal is feed to the ADC.

To quantify the 'goodness' of the sensor it is important to know approximately how the receiver circuit performs in terms of transfer function of amplifier, response time, and internal noise. The remaining part of this subsection is dedicated to analyzing the quantities.

The bandwidth of the circuit is governed by a number of factors. The first is the bandwidth of the photo detection part. The photodiode has a (parasitic) capacitance which together with the load resistor acts as a low pass filter. The time constant of this is given

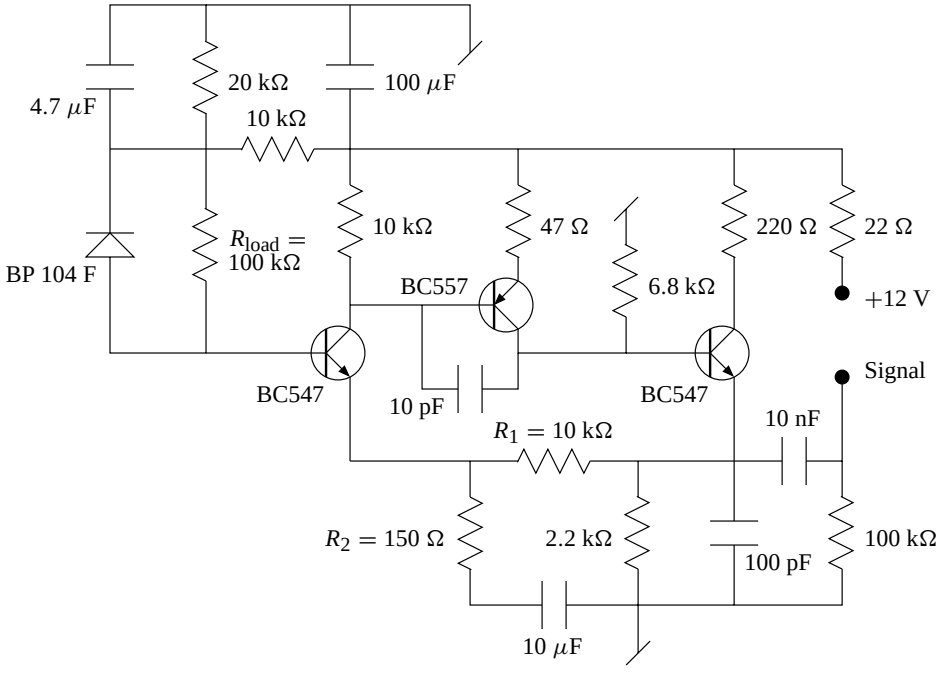


Figure 5.28: The receiver circuit used in setup 3. Source: Hyder [44]

by (4.2),

$$B = \frac{1}{2\pi \times 10^5\ \Omega \times 48 \cdot 10^{-12}\ \mathrm{F}} = 33.2\ \mathrm{kHz}\ .$$

In most sensor applications this is sufficient bandwidth, but if a higher bandwidth is needed this can obviously be achieved by using a smaller resistance.

Since the three transistors function like an op-amp the gain is given by the standard transfer equation for a non-inverting amplifier. The amplification is essentially given by R_1 and R_2, so

$$G_{\mathrm{amp}} = \frac{R_1 + R_2}{R_2} = \frac{10\ \mathrm{k}\Omega + 150\ \Omega}{150\ \Omega} = 67.7\ .$$

The bandwidth of the amplifier is more complicated to quantify, and in this context it is sufficient to state that the transfer function is band pass with break frequencies at approximately 200 Hz and 50 kHz.

To estimate the SNR in the receiver circuit it is necessary to know the approximate light power P_{r} at the receiver. The emitter was placed 3 m from the receiver and the area of the receiver is 2.2×2.2 mm , so

$$P_{\mathrm{r}} = \frac{2.2 \cdot 2.2 \cdot 10^{-6}\ \mathrm{m}^2 \times 30 \cdot 10^{-3}\ \mathrm{W/sr}}{3^2\ \mathrm{m}^2/\mathrm{sr}} = 1.61 \cdot 10^{-8}\ \mathrm{W}.$$

The current produced in the photodiode is given by (4.1), i.e.

$$i_P = \frac{1.602 \cdot 10^{-19}\ \mathrm{J} \times 0.9 \times 1.61 \cdot 10^{-8}\ \mathrm{W} \times 950 \cdot 10^{-9}\ \mathrm{m}}{6.626 \cdot 10^{-34}\ \mathrm{Js} \times 3 \cdot 10^8\ \mathrm{m/s}} = 11.1\ \mathrm{nA}\ .$$

This current generates a voltage drop over the load resistor of 11.1 nA $\times 100\ \mathrm{k}\Omega =$ 1.1 mV. This is then amplified by a factor 67.7 which yields a voltage output of 74 mV. In reality there is a spectral mismatch between emitter and receiver, resulting in an approximately 50% reduction of the transmitted light power, and the estimated voltage output is therefore 34 mV. The ADC is set to map ± 250 mV to 12 bit, so the signal generated by the diode at 3 m distance excites the 8 LSBs.

To estimate the accuracy of the CGM resulting from the transmission it is necessary to have an estimate of the noise. The external, colored disturbances have been discussed in the previous chapter, and only the internally generated noise is addressed here. The main source of noise is the photodiode and the load resistor (because of the subsequent large amplification of the noise current flowing through them). The shot noise is given by

$$i_{\mathrm{shot}} = \sqrt{2 \times 1.602 \cdot 10^{-19}\ \mathrm{J} \times 1.31 \cdot 10^6\ \mathrm{Hz} \times (11.1\ \mathrm{nA} + 2\ \mathrm{nA})} = 11.8\ \mathrm{pA}\ ,$$

where $i_{\mathrm{d}} = 2$ nA is the dark current in the diode. The thermal noise of the load resistor R_{load} is

$$i_R = \sqrt{\frac{4 \times 1.38 \cdot 10^{-23}\ \mathrm{J/K} \times 293\ \mathrm{K} \times 1.31 \cdot 10^6\ \mathrm{Hz}}{10^5\ \Omega}} = 73.3\ \mathrm{pA}\ ,$$

The total noise generated by the photo detection circuit is $i_{\text{shot}} + i_R = 85.1$ pA. This is equivalent to a voltage output at 0.6 mV which excites the 3 LSBs. The SNR the becomes

$$\text{SNR} = 20 \log_{10} \frac{11.1 \text{ nA}}{85.1 \text{ pA}} = 42.3 \text{ dB} .$$

Note that the dark current is not included in the SNR as this produces a constant offset in the current, and this is removed by the subsequent amplification.

5.5 Fourth Test Setup

The last of the four test setups is a sensor that complies with a typical commercial standard. That is, the PCB layout, the amplification, the electric shielding, and the optical and mechanical construction are all optimized. The validation in this setup will also be much more significant than in the previous setups in the sense that the error rate must be below 10^{-5}. The sensor is based on diffuse reflection like the first and second test setups. The higher the signal value is the closer the reflecting object is. Two test signal have been recorded, without and with short time disturbances. The laboratory lighting was on, but the daylight filter combined with optimized amplifier effectively removes the 100 Hz sinusoid.

A thorough examination of the 'background' noise reveals that, as in the previous three cases, the noise is normally distributed. The standard deviation is approximately 3.6 and a probability of 10^{-5} for making a false detection corresponds to a threshold of 4.3 times the standard deviation, i.e. ≈ 16. The threshold on samples classified as too small for proper detection is therefore set to $S = 8$. The sensor has a large dynamical range to handle detection in a large span of distances. The ADC saturates at signal level 205, and signal values in the entire range is expected. Therefore β must be no bigger than $1/205$

By using the $P_{\text{FP}} = P_{\text{FN}}$ curve for this particular values of β and σ (this curve is not shown) it can be determined that $P_{\text{FP}} = P_{\text{FN}} = 10^{-5}$ corresponds approximately to $\alpha = 2.1$ and $R_{\max} = 15.4$. The weakest detectable signal level is therefore $R_{\max}\sigma = 15.4 \cdot 3.6 \approx 55$.

The second validation method is now applied with these parameters to the first test signal, see to top plot in Fig. 5.29. The validation of the measurements works fine in the sense that the signal level at which more samples are classified as useful than useless is approximately 20 (this is a little difficult to see on the plot, however a magnification reveals that the level is approximately 20) while the level predicted by the theory is 16, and as the signal level increases all the samples are validated correctly.

However, there seems to be a mismatch between the theory and the present validation; the weakest detectable signal level is predicted to be 55, which corresponds rather poorly with the fact that every single sample above 35 has been classified as useful. The reason for this is that the weakest detectable signal level is the level at which the probability for making a FN decision is 10^{-5}. At lower signal levels the probability is higher, but even

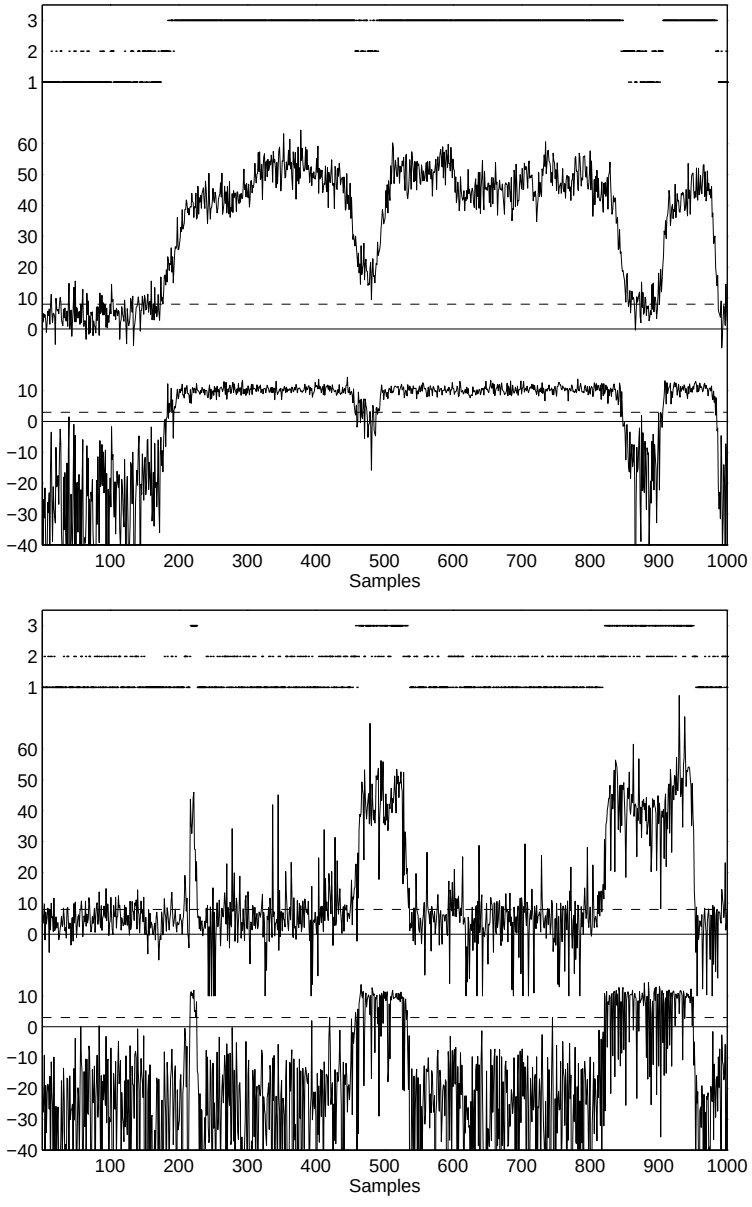


Figure 5.29: The two test signals in the fourth setup. The layout of the plots is the same as in Fig. 5.11. The top plot shows the 'clean' signal and the lowermost plot shows the signal with multiple time-localized disturbances. Here $S = 8$, $\alpha = 2.1$ (equivalent to 3.2 in dB), and $\beta = 1/205$.

a probability of, say, 10^{-3} only produces one wrong classification for every 1000 samples. And in Fig. 5.29 there are only approximately 700 useful samples. To demonstrate with test signals, and a comfortable margin for the uncertainty error, that $P_{\text{FN}} = 10^{-5}$ corresponds to a signal level of 55 requires in order of 10^7 samples.

Note that for the first validation method the weakest signal level is 32, i.e. two times the level corresponding to $P_{\text{FP}} = 10^{-5}$. The third order term in the denominator of the validation function (4.19) has as a consequence that $\tilde{\Theta}$ does not increase a fast as Θ for increasing signal level y, and thus y has to be somewhat higher to achieve the same probability.

It is particularly interesting to see the result of applying some severe external disturbance to the sensor as it is designed, mechanically as well as electrically, to be robust to such disturbances. For this purpose the remote control from in the third test setup is used. The result is the second test signal shown in the lowermost plot in Fig. 5.29. The majority of this signals is 'no object'-samples. There are three reflections, a very short one close to sample number 200, and two somewhat longer reflections. The disturbance have been applied from sample 250 and to the end of the signal.

It is obvious from the signal that a time-localized disturbance has occurred, though the transient are not nearly as powerful as in the fourth test signal in the third setup (see Fig. 5.19 on page 118). The validation of the samples responds as it is designed to do. Not a single sample from the 'no object'-parts is classified as useful, and the three reflection parts are clearly identified. But it is not easy to see whether the validation classifies all the samples in the reflection parts correctly, i.e. whether all the transients are classified as useless. However, it is reasonable to believe that they do, for the following reason. All the transients in the 'no object'-parts are classified correctly, and a signal level between 40 and 50 is classified correctly (this is evident from the first test signal). Although the validation function is not linear, this indicates that the at least the majority of the sample are classified correctly in the 'reflection'-parts.

Finally, note that while the S threshold is 8, indicating that a signal level below 8 is simply to small to be useful, setting $S = -\infty$ would not alter the number of samples classified as useful. This is obvious since the all the parts of the two signals classified as too small (dots in line 1) is also below the α threshold in the validation function $\tilde{\Theta}$. Choosing $S = -\infty$ would thus effectively result in all the dots in line 1 move to line 2. However, by doing this the distinction between 'no detection' and 'not a useful measurement' is lost.

5.6 Implementations of the Algorithm

The primary implementation of all the steps in the algorithm is in MATLAB. Some of the steps have also been implemented in C, partly to reduce the processing time in simulations, partly to test the algorithm in the real time test setups presented in this chapter. All the graphs in this chapter have been generated in MATLAB, but the signals are all recorded using the PC connected to the various test setups. In most cases the recorded signals have also been post-processed real time.

The most important steps have been implemented in C in order to make the test setup work in real time. The simplicity of the methods applied in the individual steps means that the main challenge in C is keeping track of the many indices. The C implemented steps are

- Wavelet packet transform.
- Rudin-Shapiro transform.
- Polynomial decomposition and denoising.
- Channel gain estimation.

These have all been implemented in MATLAB prior to the C implementation. The following steps have only been implemented in MATLAB, mainly due to lack of time.

- Validation of measurements.
- Transient removal and specialized test signals.
- Adaptive generation of designed signals.

All implementations in connection with the CGM algorithm have turned out to be straightforward and without any numerical stability problems whatsoever*. This is not the case for the implementation of the reflection map model presented in Chapter 8, for instance. The actual code is not printed in this thesis because it would require very many additional pages and yet only serve a minor important purposes.

The next step, and indeed the real challenge, is implementation of the CGM algorithm in signal processing hardware. This has not been attempted because it is outside the scope of this thesis. However, the suggested algorithm is prepared for a future signal processing implementation since numerical stability, low program complexity and limited dynamical range is some of the design criteria of the CGM algorithm. This has been discussed a number of times in the previous chapters. To give an idea of the stability, the rounding error after a six level wavelet packet decomposition and reconstruction in the Analog Device 16 bit fixed point processor ADSP2181 is shown in Fig. 5.30. The transform is a regular wavelet transform where the filters obey $\sum_k |h_k| = 1$ rather than the usual scaling in order to maintain an approximately fixed dynamical range. For virtually all of the samples only the 3 LSB are affected by rounding (3 LSB corresponds to 8 steps on the 'stair' in the lowermost plot).

For comparison the dynamics in an implementation of a discrete cosine transform in shown in Fig. 5.31. Thanks are due to Jan Østergaard for doing the necessary computations in order to make this figure.

*The conditioning matrices used in the moment preserving edge filter wavelet transform did present a significant stability problem. However, this is not a implementational issue, but a intrinsic problem in the edge filter construction.

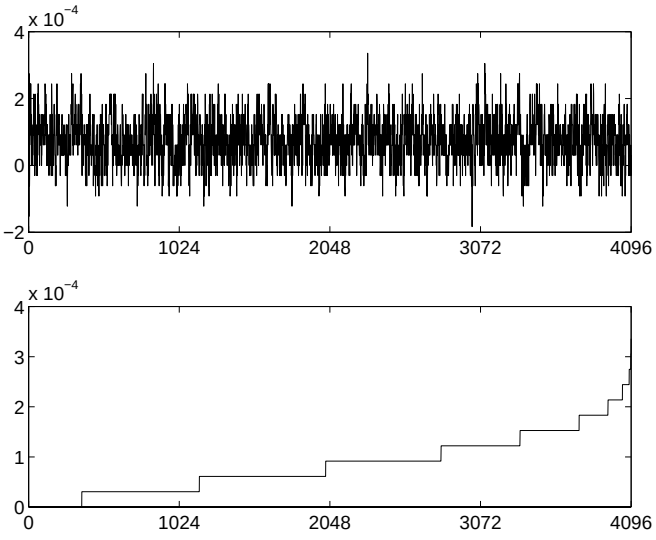


Figure 5.30: The rounding error in a six level wavelet packet decomposition and reconstruction of a length 4096 music signal in the 16 bit fixed point processor ADSP2181. The lowermost graph shows the error signal sorted according to amplitude.

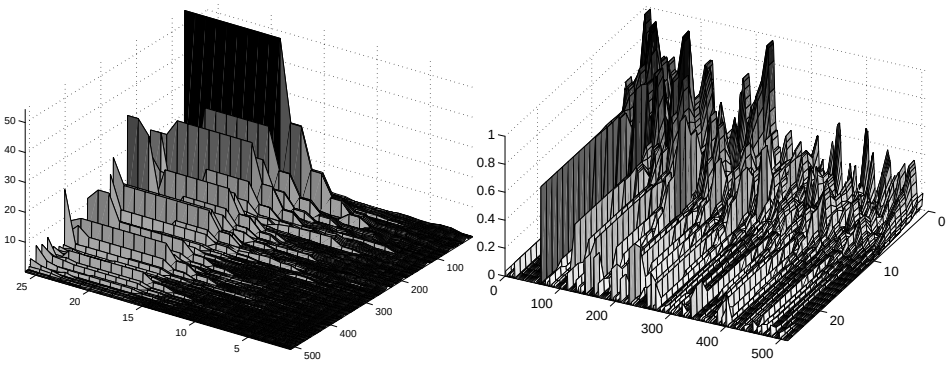


Figure 5.31: The dynamics throughout the 26 butterfly and scaling steps in a direct implementation (left) and tuned implementation (right) of a 512 DCT applied to the same music signal as the WPT in Fig. 5.30.

5.7 Conclusion

The main purpose of Chapter 5 is to demonstrate the various methods and ideas presented in Chapter 4. This is done by recording a series of test signals by means of four test setups. While all the setups use infrared technology for emission and reception the electric circuit varies in design, and thus a range of different principles for handling the electric signals have been tested.

All the major steps in the CGM algorithms are tested in the four test setups. These are wavelet and RS modulated transmission signals, identification and removal of time and frequency-localized noise, and two methods for validation of measurements. In general, the suggested algorithm works quite well and fulfills the expectations presented in Chapter 3. In particular, it has been demonstrated that by combining denoising and validation of measurements it is possible to achieve a significant level of robustness in the presence of noise. At the same time the computational load and complexity of the solutions are acceptable, and there are no numerical stability issues. Both of these results are important for implementation in low-cost signal processing hardware.

implementations in C has shown that the wavelet transform as well as the Rudin-Shapiro transform have a low program complexity, and are easily portable from one programming environment to another. The C implementations also provide good estimates of the quantified computational load (this is a matter of counting operations in the code).

Part II

Spatial Position

Methods for Determining Spatial Position

6

A fast, robust, and inexpensive determination of the three dimensional position of a passive object is an interesting scientific challenge. It is also an interesting functionality from an industrial and commercial point of view. There are a series of technical and theoretical challenges which must be overcome before this functionality can be achieved in small and low-cost sensors. This part of the thesis focuses on generic methods for providing this functionality. More accurately, two of the more important theoretical aspects are presented and discussed. The author have chosen to delimit this part of the thesis to only those two aspects since the subject of spatial position sensors is simply too extensive for a proper treatment in the present context. Thus, the author does not claim to have even remotely overcome the challenge of constructing a small and low-cost 3D sensors, but rather to have provided a valuable input to the process of designing and constructing the 3D sensor.

6.1 Introduction

As opposed to a system determining 3D positioning of active objects, (such systems are well-known and widely used; one of many examples is the global position system, GPS), a system for determining position of passive objects usually has to rely on signals which are emitted in the direction of an object and reflected by the object, rather than signals emitted by the object itself. This approach poses two basic challenges. Emitting and receiving a signal in order to obtain information about the object, and converting this information into a spatial position. The first challenge was faced in the first part of this thesis with a CGM as the result. This part is therefore dedicated to converting CGMs, i.e. intensity of reflections, into a spatial position.

To make the challenge more tangible it is useful to have a specific application in mind. The author suggests an infrared touch-free 3D mouse. Many other positioning systems could be used, but the 3D mouse has been chosen because it is cheap and relatively easy to build (once the theory is ready to be put to a test), has suitable real time requirements, is of some commercial interest, and it has ‘laboratory-friendly’ dimensions. The idea is to emit a whole range of signals from various position and measure the reflected intensities. The relations between the intensities is then converted into a spatial position. The signals

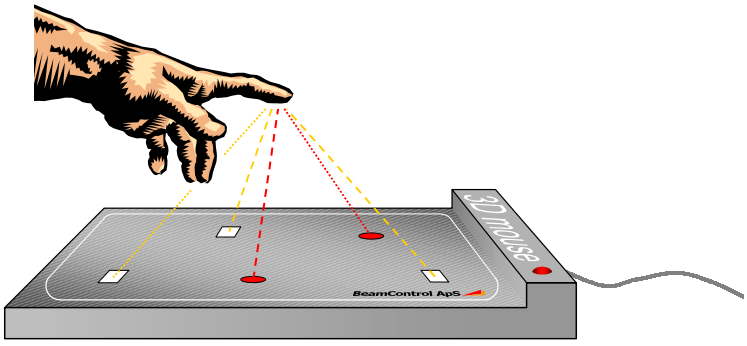


Figure 6.1: The physical design and basic principle of the 3D mouse. A number of infrared emitters (circles) and receivers (squares) are located in the 3D mouse under the IR transparent top cover. All necessary electronics are also located inside the 3D mouse.

are infrared light, and the IR emitters and receivers are located in a box with dimensions equivalent to a thick ordinary mouse pad. The spatial position of an object, like a hand, can then be determined when it is over the mouse pad. The physical design and basic principle of the 3D mouse is shown in Fig. 6.1.

6.2 Determining the Spatial Position

The basic mathematical problem in determining the spatial position is mapping a high dimensional data vector, which is noisy measurements made on an object in space, into a three dimensional data vector containing the coordinates of the object. The challenge is not to define or create the mapping per se, but to devise a method that produces a fairly accurate result when noise is present in the CGM.

There are a number of a priori feasible ways for converting the CGMs, ranging from purely analytical derived equations to a table of a discretized mapping based on meticulously measured CGMs. The former is definitely preferable to the latter since analytical approach allows for parameterization of the mapping. This would be quite useful in applications where the conditions are likely to change. The question is to what extent it is reasonable to rely on the modeling needed for an analytical solution.

In any case the mapping must fulfill some basic requirements to be usable in real applications. The following prioritized list shows these requirements. The mapping

1. works well for good measurements,
2. yields a reasonable relation between error in measurements and error in 3D position,
3. has low computational complexity,
4. has low dynamic range in computations,
5. is easily adaptable in real time,

Since the measurements are expected to be good most of the time, the primary concern is that the mapping does well in this case, and the second requirement ensures that a small decrease in accuracy does not result in too large deviations in the spatial position.

In an attempt to construct a mapping by theoretical means a geometrical description is presented in the following chapter. A neural network solutions has also been investigated, although not nearly to the same extent. This latter solution in presented in Section 6.3.

6.2.1 Geometrical Approach

The mapping of high dimensional data to three dimensional position can, theoretically, be accomplished by purely geometrical consideration. Such a solution is desirable since it gives a thorough understanding of the properties of a 3D setup, and it provides formulas for the dependencies and correlations that exists between the physical and electrical components in the setup. This approach requires models of the individual physical components in the setup, which in turn requires a series of assumptions and approximations. In experience the use of geometry in combination with approximations of various kinds is challenging, because geometric equations tend to be numerically sensitive. The fact that the CGMs usually originates in signals with a low SNR only makes the geometric solution even more challenging. To demonstrate the usefulness (or lack of same) a rather naive geometrical solution to the mapping problem is presented in Chapter 7.

6.3 Neural Network

There are various ways of constructing this mapping ranging from completely theoretical, geometrical consideration to purely ad hoc methods. This section present a choice which adopts the best of those two extremes (that is the intention, anyway). On the one hand a neural network offers a systematic and fairly well-described way of defining and describing the desired mapping, and at the other hand requires a lot of guessing and testing. Moreover, as is shown in the following a neural net has the potential of fulfilling the requirements mentioned in the previous section.

In this particular framework there are two ways of using a neural net; as a classifier and as a function approximation. The former is useful when only one of a few possible positions are needed instead of the actual position. This applies, for instance, when the hand is pointing at icons on a monitor. In this presentation only the function approximation network is investigated, being the most interesting type in the case of the 3D mouse.

A radial basis function (Gaussian) network has been chosen because it is well-suited for function approximation, plus it requires only a relatively limited amount of training. For a more detailed description on radial basis function network, see Chen et al. [15].

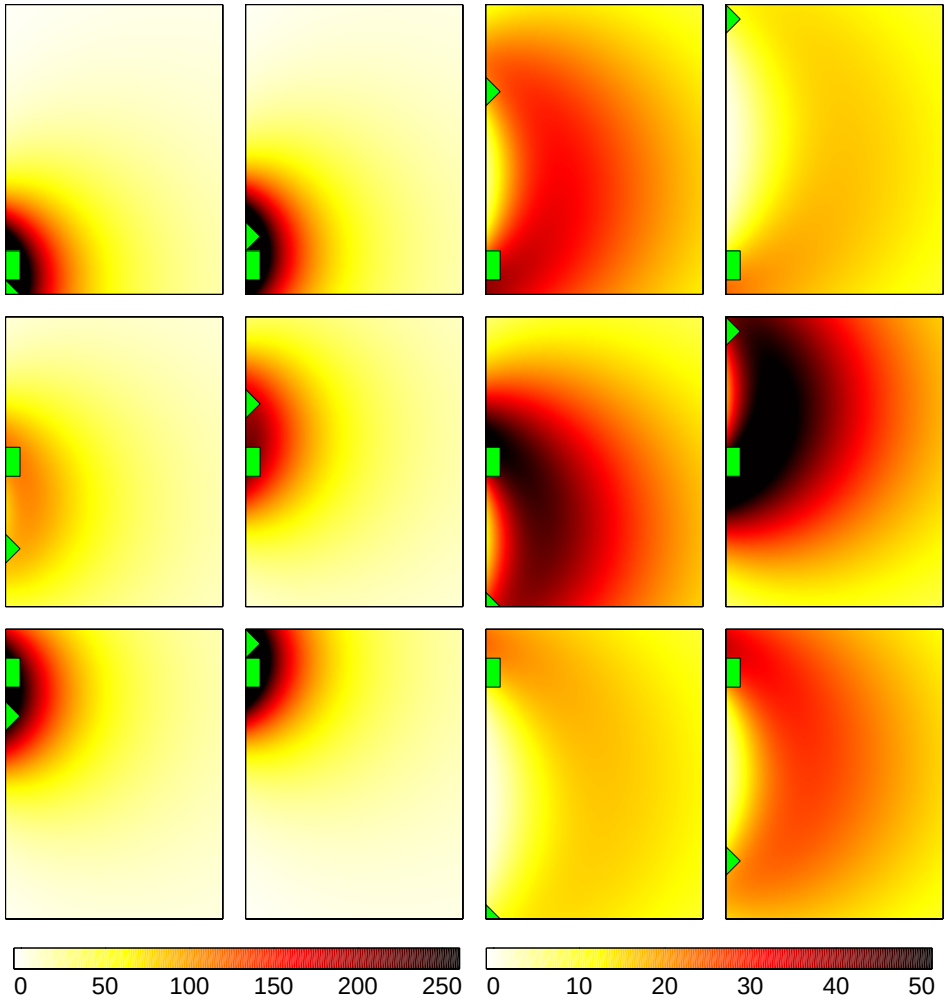


Figure 6.2: The simulated reflection intensities. The triangle is the emitter, and the square the receiver. The first two columns have the same color scale, and so does the last two. The axes limits are the same as in the other figures.

6.3.1 Simulating the Spatial Positioning

To estimate the spatial position by a neural net it is necessary with measurement data from an array of emitters and receivers. Although a data set has been acquired (it is presented in Chapter 8) this preliminary examination relies on modeled reflection maps. This is because real measurements are certain to be erroneous, simply because of the physics (light emission and reception is a quantum mechanical process subject to uncertainty). To have complete control of the measurements, and to eliminate noise in this first experiment, a model is used to produce the measurements. This is a rather complex and computationally heavy model, which simulates the reflection of a sphere at any given position by means of a ray-tracing like procedure. The model produces data close to equivalent real measurements. The model is presented in detail in Chapter 8. The model is in two dimensions for reasons discussed in the chapter presenting the model. One of those reasons is that it is easier to test the various ideas for creating mappings when it is done in two dimension. The philosophy is that if the mapping does not work in two dimensions it will not work in three dimensions.

The setup simulated here consists of 4 emitters and 3 receivers located along a line. A line represents a reduction of the dimensionality by one, and thus is equivalent to placing the emitters and receiver in a plane in the case of the 3D mouse. The simulated measurements are shown in Fig. 6.2 along with the location of the emitters and receivers. Note that the overall intensity depends heavily on the distance between emitter and receiver. The size of each measurement set is 16×20 units (which might be interpreted as centimeters). The idea is now to use a neural net to map the 12 dimensional measurements to a spatial position. The network is constructed by repeatedly adding neurons (which in this case are functions on the form $Ae^{-x^2-y^2}$) until the mean square error (MSE) between the true and simulated 2D position in a set of training points is below a threshold. Two nets have been trained, one with 8×9 training points, and one with 13×13 training points. Experiments have shown that a slightly uneven distribution of training points yields a better approximation. The training points and the accuracy of the two nets are shown in Fig. 6.3. The MSE of all the training points are 0.3 units.

The neural network is here trying to 'guess' the position of the reflection object by means of the CGMs. The net has been trained by providing the true set of CGMs corresponding to each point in 2D, and the net is then approximating the mapping from CGMs to 2D by Gaussian functions. The shading of the plots shows how wrong, in the plot units, the neural network predicts the 2D position based on the true CGMs. The net is not completely accurate because a limited number of neurons, Gaussian functions, have been used for the approximating mapping. The net corresponding to the left plot in Fig. 6.3 has 57 neurons, while the net corresponding to the right plot has 82 neurons.

While this procedure specifically reduces the error in certain points, the goal is to have a good approximation in all 2D points. In between the training points, however, there is no control of the the error, which can easily become quite large. But adding training points in places with large error will inevitably also increase the number of neurons (to meet the MSE threshold requirement).

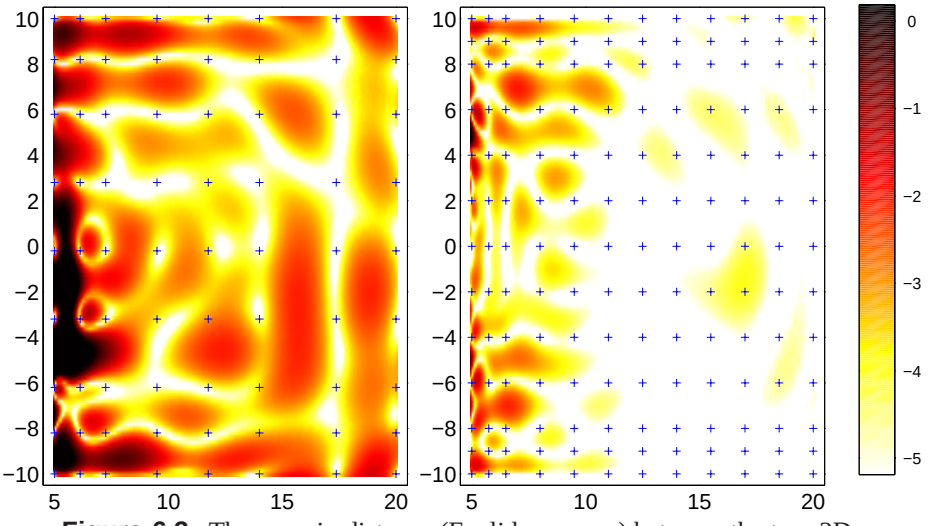


Figure 6.3: The error in distance (Euclidean norm) between the true 2D positions and 2D positions simulated by a neural net. The crosses mark the training points. The left set of training points yielded 57 neurons, while the right gave 82 neurons. The color scale is $\log_2$.

One obvious goal is to have as few neurons as possible, but it is equally important that the neural net is not too sensitive to noise. To test this (on the neural net with 82 neurons) the net has predicted the 2D position based on the 12 dimensional CGMs with added Gaussian noise. This is shown in Fig. 6.4. Here the shading also shows the deviance of the predicted 2D position from the true 2D position. Since the added noise have the same variance all over the 16×20 plane, while the measurements are varying in amplitude (as seen in Fig. 6.2), the SNR varies somewhat. Although the weaker noise is typical for laboratory tests (an SNR of 42 dB was estimated in one of test setups in Chapter 5), the more powerful noise is not uncommon. Fig. 6.4 shows one major weakness of the neural net; the large sensitivity to even Gaussian noise. It is obvious that the predicted positions are useless.

The problem is that although the 'clean' measurements are 12 dimensional, they constitute a 3 dimensional sub-manifold since they are originally mapped from $\mathbb{R}^3$. If a 12 dimensional measurement is too far from this embedded sub-manifold the prediction made by the neural net becomes arbitrary and hence useless. There at least two potential solutions to this; the neural net could be trained with erroneous data as well, or some projection onto the three dimensional sub-manifold could be applied in the 12 dimensional data space. The former idea has been tested with a partly positive result, but a significantly larger number of neurons is needed, since an even more complicated structure than the three dimensional sub-manifold is approximated. The latter solution is somewhat more

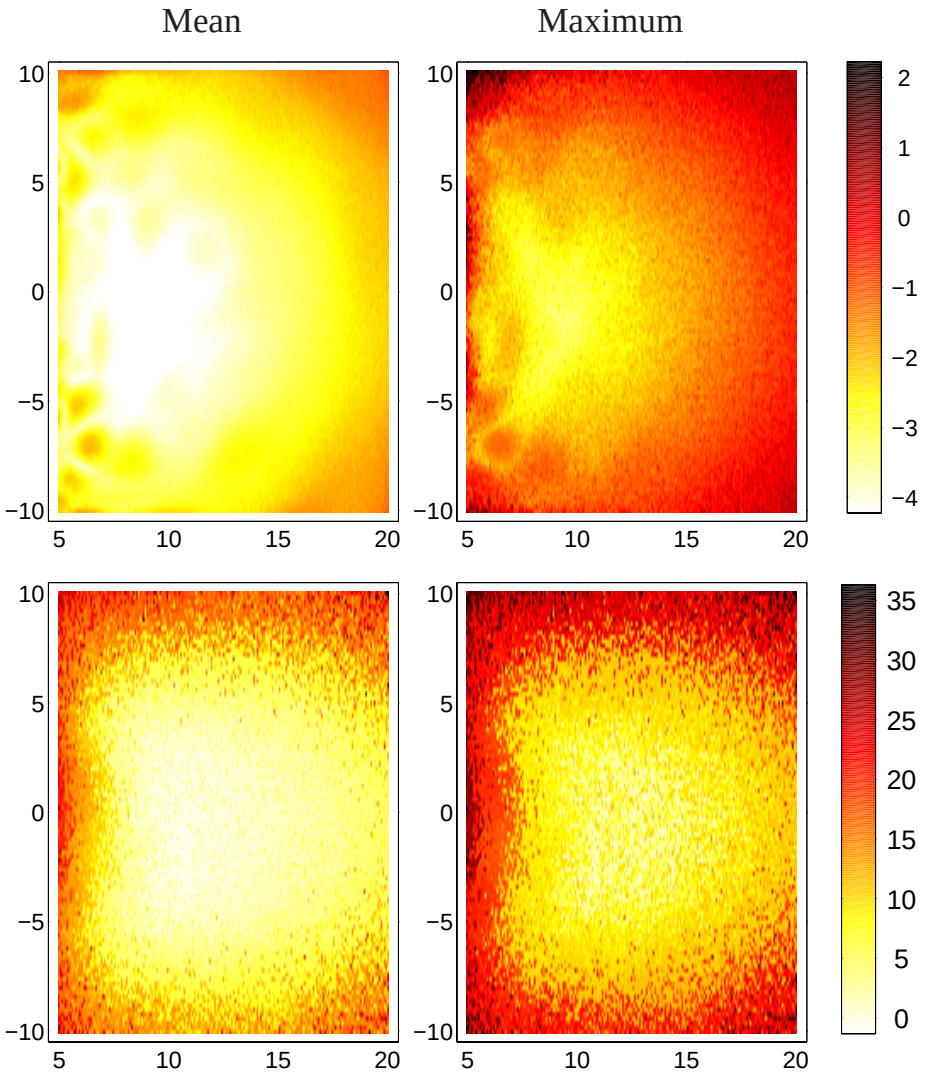


Figure 6.4: The mean and maximum distance error for 200 instances of Gaussian noise (SNR range from 50 to 25 dB in the uppermost row, and 30 to 5 dB in the lower row). The shading shows for any given 2D position how wrong, in plot units, the neural network predicted 2D position becomes when noise is added to the CGMs. The color scale is $\log_2$.

complicated because it requires a fine-gridded non-Euclidean multi-dimensional structure (consisting of splines, for instance) of the three dimensional sub-manifold in order to facilitate computation of a numerical projection, as an analytical projection is not expected to be feasible. This idea has not been tested in this thesis.

6.4 Future Work on Spatial Position Sensors

This Part of the thesis presents two different approaches to implement the mapping of CGMs to spatial position, and a measurement of the reflection characteristics in a setup with infrared emitters and receiver. These contributions might be useful in the design of a spatial position sensor, but they do not themselves represent a research effort sufficient for actual constructing a spatial position sensor. This section briefly reviews some of the work which the author consider as necessary, but has not been addressed in this thesis.

The primary task is to determine what method is appropriate for creating the mapping from measurements to position. Two methods which a priori seemed promising have been suggested. A neural network is briefly examined, and a geometrical approach is investigated more thoroughly.

The result of the examination of the neural network was inconclusive. The brute force construction of the network did not yield satisfying results. However, a number of solutions were suggested, one of which has actually proven fairly good. But the complexity of this solutions is quite high, and it has not been tested and simulated to an extent which justifies any final conclusion to be drawn. Much more work is needed to determine the feasibility of a neural network generated mapping.

The investigation of a geometrical approach is also in some sense inconclusive. While the outcome of the investigation is a series of interesting and probably useful observations, it fails to finally conclude on whether this approach is doable. Comparing the obtained result to a real measurements the geometrical modeling turns out to be inadequate under the given assumptions, and it is an open question if a more complex model, or indeed an adaption of the physical setup to the assumptions, would do the trick. Consequently, it is necessary to further develop the geometrical model if this approach is to be used for the mapping.

To get a good feeling with a suggested solution, i.e. to discover its advantages and disadvantages, and to estimate its potential, it is imperative to make extensive testing with real data from a real setup, either in real time or off-line. This also lacks from the present research effort.

A measurement of the reflection characteristics of a real setup with one emitter and one receiver has been made, however. This is not done to provide the above mentioned test data. They have been recorded partly to estimate the how realistic the model assumptions are, and partly to see if it is possible to model, and possible parameterize, the reflection characteristic. While it is possible to model the characteristics fairly well, the complexity of this modeling prohibits any easy parameterization.

Geometric Solution based on Intersections of Spheroids

7

The basic concept in the geometric approach is to derive the mapping from high dimensional measurement data to 3D position by purely analytic means. In order to do this the entire setup is modeled geometrically. The assumptions used for this model leads to a description of the mapping by a set of equations describing intersection curves for three dimensional prolate spheroids, that is ellipses revolved around their semimajor axes. This chapter presents the making of the mapping. This includes a more detailed description of the concept and assumptions leading to spheroids, a rigorous derivation of the intersection of said spheroids, a discussion of the choice of locations of emitters and receivers, a discussion of the usefulness of the model, and a series of examples to demonstrate various concepts presented throughout this chapter.

7.1 The Basic Concept of a Geometrical Solution

The construction starts with the observation that an emitter/receiver (E/R) pair transmits an 'amount' of light from emitter to receiver. This amount depends on various factor such as directional characteristics of the E/R pair and the position and properties of the reflecting object. This means that there is a vector function M mapping 3D position in front of the emitter/receiver pair to an intensity. When an intensity I is measured at the receiver it is immediately known that the reflecting object is located somewhere in the isocandela set corresponding to I of this mapping, since the isocandela set is the set of points in 3D which – according to the mapping – yields a reflected intensity I. When sufficiently many such 3D sets (each set stems from a particular emitter/receiver pair) are known for the same object, the intersection of these sets is a single point, which is then the location of the object. What we want eventually is the 'reverse' mapping $\mathcal{U}$, called the intersection function, which given a set of intensities (that is the CGMs) provides the intersection point for the isocandela sets corresponding to those intensities, thus yielding the 3D position of the reflecting object.

In this chapter the reflecting object is assumed to have properties (see the next section) such that this mapping $M : \mathbb{R}^3 \mapsto \mathbb{R}$ is on the form

$$M(\mathbf{p}) = \big(\|\mathbf{p} - \mathbf{e}\| + \|\mathbf{p} - \mathbf{r}\|\big)^{-2} , \tag{7.1}$$

where $\mathbf{p}$, $\mathbf{e}$, and $\mathbf{r}$ are the 3D position of the object, the emitter, and the receiver, respectively. The locus of $M(\mathbf{p}) =$ constant is easily seen to be a prolate spheroid with focal points in $\mathbf{e}$ and $\mathbf{r}$. Note that this observation is independent of the power -2, which is included here merely for visual reasons to model the reduction of the intensity by the square of the distance to the object. This does in fact not have any qualitative influence on the final mapping function $\mathcal{U}$.

Note that the form (7.1) requires the object to reflect the light without any scattering. The $+$ sign holds the implicit assumption that the light 'continues on' when reflected instead of being scattered. If the object scatters the light the intensity function would be along the lines of the form

$$M(\mathbf{p}) = \left(\|\mathbf{p}-\mathbf{e}\|\|\mathbf{p}-\mathbf{r}\|\right)^{-2}, \tag{7.2}$$

that is with multiplication rather than addition. The reflection map is very different from that of (7.1), which is evident by inspection, see Fig. 7.10 on page 171. The form (7.2) is admittedly strongly simplified, but it does show the basic form when two scatterings are involved (the first 'scattering' happens as the emitter emits light in many directions). Thus, the model (7.1) requires the object surface to be mirror-like.

Obviously, in the real case the mapping M depends on the orientation and reflectivity of the object, and these properties should therefore be included in the mapping function (which they are clearly not). This also goes for other properties such as emitter and receiver characteristics etc. However, even simple assumptions prove difficult to accommodate in the mapping, and will therefore not be included in this geometric solution. Furthermore, the idea of analytically deriving the intersection function for the isocandela sets does not depend in a fundamental way on the aforementioned properties (the complexity of the approach does to a very high degree, though). Consequently, the assumptions are chosen such that they are very simple and results in a relatively simple intersection function $\mathcal{U}$. To what extent the resulting model and intersection function mimic the real world is a question still to be answered (however, see the discussion of the geometric approach in Section 7.5 and 7.6 starting on page 170).

Although the results presented in this chapter are mainly of analytical nature there are no (to the best of the authors knowledge) literature to support them (except in a few peripheral cases, which are marked with citations).

7.2 Assumptions

In a real setup of E/Rs there is a series of factor which should be accounted for. Since most of these factors add significantly to the complexity, a sufficient model might be very complicated and difficult to handle. In this chapter most of these factors have been left out in order to simplify the model to an extent which allows for relatively simple equations. The factors in question are

Directional characteristic of the E/Rs. It is most common that emitter has a directional characteristic which is not just significantly different from being uniform, but also asymmetric and uneven. Although the receivers are also not uniform, they do usually exhibit a nice characteristic, such as a cosine. In this model both characteristics are assumed to be uniform.

Noise The ever present problem of noise has not been accounted for in this model. This means that there has been no attempt to robustify the equations (for instance by adding some kind of low pass filtering property). There is currently no guarantee that even small perturbations of the high dimensional data will be handle properly. One analytical steps has been taken to reduce the influence of noise, however. This is related to the location of the sensors.

Location of E/Rs Since it is a priori unknown what the optimal locations of emitters and receivers are it is desirable to have complete freedom. The question of the optimal locations of E/Rs is discussed in Section 7.4. In this model there is one requirement, however. The emitters and receivers have to be located such that each emitter is located adjacent to a receiver. Such an E/R pair is in this chapter referred to as a sensor. Note that it still makes sense to talk about an E/R pair, which is an emitter and a receiver that are not necessarily located adjacently.

Characteristics of reflecting object The most 'unrealistic' assumption in this model is the characteristics of the reflecting object. To reduce the complexity it is assumed that the object is reflecting the light in such a way that the 3D isocandela map of an E/R pair consists of concentric spheroids. This assumption holds only when the received intensity is related only to the distance from the emitter to the object and back to the receiver as shown in (7.1). This in turn is true only when the object reflects light without any scattering and in the direction of the receiver (a reduction of the intensity is allowed). This can be achieved for instance by a plane mirror positioned such that it is tangent to the spheroid with focal points in the emitter and the receiver, or by a sphere mirror. At the same time the reflection has to happen at the same point in 3D for all involved pairs of E/Rs (to justify the idea of a spheroid intersection point). The obvious conflict in 'the spheroid assumption' (that is, having reflection in different direction at the same point) raises the question of to what extent this assumption is valid in any real setup. To investigate possible solutions to this problem the characteristics of the reflecting object is discussed in detailed in Section 7.5.2. The possibly unrealistic assumptions does make the geometric equations relatively simple, however, and thus allows for a not too complicated analytical solution to the problem of mapping high dimensional measurement data to 3D.

Note that the reflectivity of the object is accounted for in the model, since it includes a uniform scaling r of all the intensity measurements.

7.3 The Intersection Function

The smallest number of spheroids with which it is possible to uniquely determine an intersection point is three. The smallest number of sensors that gives three spheroid is also three. From this set of three sensors any two give the focal points of a spheroid. The objective of this section is to construct the function which maps these three spheroids with different semimajor axes given by the three measured intensities into that particular point in 3D where they intersect. The measured intensities are not used 'as is' since the transmitted light is subjected to a unknown reduction (governed by the reflectivity of the object and amplification in the receiver). To account for this the intensities are all scaled by a common factor r. A more detailed description of this scaling is given in Section 7.3.4 on page 155. Note also that throughout this chapter the focal points are in the xy plane.

7.3.1 Definitions

Before deriving the intersection function it is convenient to reduce the number of degrees of freedom to a minimum by means of scaling, rotation, and translation of the triangles spanned by the sensors. In order to do this a few new variables are needed. They are introduced in Definition 7.2. First of all, the rotation operation needs to be defined (scaling and translation is trivial).

Definition 7.1 (Givens rotation)
A clockwise rotation θ radians of a point in xyz space around the z axis is accomplished by multiplying with

$$\mathbf{G}(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Definition 7.2 (Focal Points)
Let P, Q, and S be three points in the xy plane. Define the vectors

$$\mathbf{p} = \begin{bmatrix} Q_1 - S_1 \\ Q_2 - S_2 \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} P_1 - S_1 \\ P_2 - S_2 \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} Q_1 - P_1 \\ Q_2 - P_2 \end{bmatrix},$$

define the angle

$$\theta = \frac{q_2}{|q_2|} \arccos\Big(\frac{q_1}{\|\mathbf{q}\|}\Big),$$

and define $\mathbf{d}$ *and* γ *as*

$$\mathbf{G}(\theta) \begin{bmatrix} \mathbf{q} & \mathbf{p} \\ 0 & 0 \end{bmatrix} = 2\gamma \begin{bmatrix} d_1 & d_2 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

The points PQS must be such that

$$\langle \mathbf{s} \times \mathbf{q}, \mathbf{e}_3 \rangle > 0 \qquad \textit{and} \qquad \langle \mathbf{p}, \mathbf{q} \rangle < \|\mathbf{p}\|\|\mathbf{q}\|,$$

and $d_1 \geq d_2 \geq 0$.

The definition is closely related to Fig. 7.1.

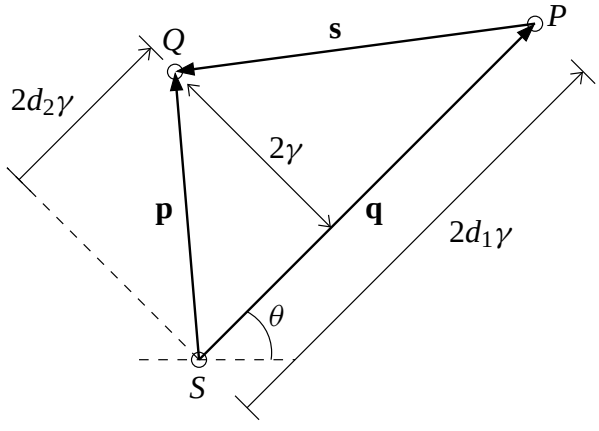


Figure 7.1: The setup for the focal points.

Note that any set of three points in the xy plane complies with this definition when

(i) they are not lying on a line,
(ii) they are enumerate with P, Q, and S counter-clockwise, and
(iii) Q is associated with the obtuse angle (whenever there is one).

Thus, effectively, this includes all triangles.

The next step is to introduce the spheroids. According to the assumptions presented in the beginning of the chapter they are prolate spheroids, that is they are given as the locus of a revolution of an ellipse around it's semimajor axis.

Definition 7.3 (Prolate Spheriod I)
Define the prolate spheroid form by

$$E(\mathbf{c}, \mathbf{r}, \theta) = \mathbf{c}^\top \mathbf{G}(\theta) \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_2 \end{bmatrix}^{-1} \mathbf{G}(-\theta)\mathbf{c}.$$

It is not immediately obvious how this definition relates to a prolate spheroid with focal points in the xy plane. The prolate spheroid form is obtained by first noting that the equation

$$\frac{(x - x_0)^2}{r_1^2} + \frac{(y - y_0)^2}{r_2^2} = 1$$

is an ellipse with centre in (x_0, y_0), semimajor axis r_1 parallel to the x axis, and semiminor axis r_2. Then the following lemma justifies the definition.

Lemma 7.4 (Revolution of Ellipse)
A spheroid described by revolving the ellipse

$$\frac{(x-x_0)^2}{r_1^2}+\frac{(y-y_0)^2}{r_2^2}=1 \tag{7.3}$$

around the line $\begin{bmatrix} x_0+t & y_0 & 0 \end{bmatrix}$ *followed by a rotation* θ *clockwise around the line* $\begin{bmatrix} x_0 & y_0 & t \end{bmatrix}$*, is given by*

$$E\left(\begin{bmatrix} x-x_0 \\ y-y_0 \\ z \end{bmatrix}, \begin{bmatrix} r_1^2 \\ r_2^2 \end{bmatrix}, \theta\right)=1\,. \tag{7.4}$$

Proof
Assume without loss of generality that $x_0=y_0=0$. Expanding (7.4) then yields

$$\frac{\big(x\cos\theta-y\sin\theta\big)^2}{r_1^2}+\frac{\big(x\sin\theta+y\cos\theta\big)^2}{r_2^2}+\frac{z^2}{r_2^2}=1. \tag{7.5}$$

Define the result of a counter-clockwise rotation of the locus of (7.5) around the z axis as $\begin{bmatrix} \tilde{x} & \tilde{y} & 0 \end{bmatrix}^\top = \mathbf{G}(-\theta)\begin{bmatrix} x & y & 0 \end{bmatrix}^\top$. Then

$$\frac{\tilde{x}^2}{r_1^2}+\frac{\tilde{y}^2}{r_2^2}+\frac{z^2}{r_2^2}=1,$$

which is the result of revolving the ellipse (7.3) around the x axis. □

7.3.2 Examples

Throughout this chapter a number of examples will be given to support the various theorems, lemmas etc. They are all based on the same setup which is presented here.

A total of four sensors are located in the xy plane at $F_1 : (3,2)$, $F_2 : (9,5)$, $F_3 : (6,10)$, and $F_4 : (2,9)$, see Fig. 7.2. This generates four triangles $F_1F_2F_3$, $F_2F_3F_4$, $F_3F_4F_1$, and $F_4F_1F_2$ (which all complies with Definition 7.2). The reflecting object is located in $F_5 : (5,4,3)$. This point gives a set of measurements which are related to the distances from the sensors to the point (i.e. the object). While measured reflected intensities are proportional to the square of reciprocal of the distance, the 'measurements' used in the following equations are assumed to be proportional to the distance. Thus, in a real application it is necessary to apply a mapping on the form $(\cdot)^{-2}$ to the measured intensities. Note that the mapped measurements are also referred to as measurements.

There are two equal measurements for each sensor pair. In a real setup the two measurements will most likely not be equal due to noise, and the redundancy can the be used to decrease the noise level. Since there are a total of six different combinations of two

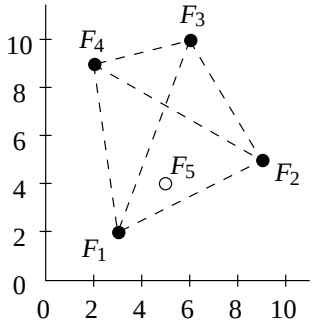


Figure 7.2: The locations of the four sensors and the projection onto the xy plane of the location of the object.

Table 7.1: The values according to Definition 7.2 for the four triangle.

Triangle	θ	θ	γ	γ	d_1	d_2
$F_1F_2F_3$	$\arccos\left(\frac{3}{\sqrt{73}}\right)-\pi$	-1.9296	$\frac{39}{73\sqrt{73}}$	2.2823	$\frac{73}{39}$	$\frac{31}{39}$
$F_2F_3F_4$	$-\arccos\left(\frac{7}{\sqrt{65}}\right)$	-0.51915	$\frac{23}{65\sqrt{65}}$	1.4264	$\frac{65}{23}$	$\frac{3}{4}$
$F_3F_4F_1$	$\arccos\left(\frac{3}{\sqrt{73}}\right)$	1.2120	$\frac{29}{73\sqrt{73}}$	1.6971	$\frac{73}{29}$	$\frac{53}{29}$
$F_4F_1F_2$	$\pi-\arccos\left(\frac{7}{\sqrt{65}}\right)$	2.6224	$\frac{45}{65\sqrt{65}}$	2.7908	$\frac{13}{9}$	$\frac{2}{3}$

sensors there are also six measurements. Simulated measurements corresponding to the point F_5 are given in Table 7.2. Here the measurements are actually the sum of the distance from the object to the two sensors, and not intensities.

Table 7.2: Simulated measurements for the point F_5 (see Fig. 7.2).

Sensor pair	F_1F_2	F_1F_3	F_1F_4	F_2F_3	F_2F_4	F_3F_4
Measurement	9.2221	10.905	10.681	11.881	11.656	13.340

7.3.3 Fixing the Focal Points

The general intersection function is based on the intersection function for spheroids with fixed focal points. Actually, the section title is slightly misleading since they are not all fixed, but the restrictions imposed on them reduce the number of degrees of freedom from six (three points times two dimension) to three. This simplifies the construction somewhat and the remaining degrees of freedom are easily introduced again later.

Since each set of three spheroids will generate exactly one intersection function the following derivations are, unless otherwise stated, for three spheroids and their three focal points.

The three focal points are denoted P, Q, and S (see also Fig. 7.1). The point S is fixed in the origin $(0, 0)$, the point P is constrained to the x axis, that is $P = (2d_1, 0)$ with $d_1 > 0$, and Q is constrained to the horizontal line $x = 2$, i.e. $Q = (2d_2, 2)$. Furthermore, Q is always the obtuse angle, so we also have $0 \leq d_2 \leq d_1$. This is equivalent to Definition 7.2 with $\theta = 0$ and $\gamma = 1$. To avoid symmetry one more restriction could be imposed (for instance $2d_2 < d_1$). However, this 'redundancy' does not complicate the following computations (they are actually a little easier without this restriction), and moreover, it does not reduce the number of degrees of freedom.

The three spheroids generated by the three focal points each have one degree of freedom, namely one of their semi axes. In this setup the semimajor axes are free. The measurements made in the physical setup are the intensities of the reflected and received light. The results of the $(\cdot)^{-2}$ conversion are denoted w_1, w_2, and w_3, and corresponds to the measurements made for $|PQ|$, $|QS|$, and $|PS|$, respectively. That is, a w_k is proportional to the distance from an emitter to the object plus the distance from the object to a receiver. The assumption that all emitters have the same uniform characteristics (and ditto for the receivers) leads to a single unknown variable r, which represents the level or amplitude of these characteristics.

In the following equations describing the three spheroids the semimajor axes are $w_n r$, and the semiminor axes are computed based on the fact that the square of the semimajor axis equals the square of the distance between focal points minus the square of the semiminor axis.

$$PQ: \quad E\left(\begin{bmatrix} x - d_1 - d_2 \\ y - 1 \\ z \end{bmatrix}, \begin{bmatrix} w_1^2 r^2 \\ w_1^2 r^2 - (d_1 - d_2)^2 - 1 \end{bmatrix}, \arctan\left((d_1 - d_2)^{-1}\right)\right) = 1 \tag{7.6}$$

$$QS: \quad E\left(\begin{bmatrix} x - d_2 \\ y - 1 \\ z \end{bmatrix}, \begin{bmatrix} w_2^2 r^2 \\ w_2^2 r^2 - d_2^2 - 1 \end{bmatrix}, -\arctan d_2^{-1}\right) = 1 \tag{7.7}$$

$$PS: \quad E\left(\begin{bmatrix} x - d_1 \\ y \\ z \end{bmatrix}, \begin{bmatrix} w_3^2 r^2 \\ w_3^2 r^2 - d_1^2 \end{bmatrix}, 0\right) = 1 \tag{7.8}$$

All the expressions in the above equations are easily derived from geometrical observations using the triangle in Fig. 7.1.

The purpose of this Section 7.3 is to demonstrate that for all values fixed such three spheroids have at most one intersection point in $\mathbb{R}^2 \otimes \mathbb{R}^+$ ($\mathbb{R}^+$ is the non-negative half-line), and that the locus of the intersection for variable r is a well-defined and well-behaved curve. Further, it is demonstrated that the same holds for the more general case with arbitrary focal points. This is done in several steps, starting in the next section with an exemplification of the restriction introduced previously. This is followed in

Section 7.3.5 by the derivation of the intersection function in the restricted case. Finally, the general case is treated in Section 7.3.6.

7.3.4 One Embodiment of the Spheroids

Assume that a reflecting object has been positioned in F_5 and that the measurements given in Table 7.2 have been obtained. Using (7.6) through (7.8) three spheroids can be constructed, each corresponding to a set of two corners in, say, the triangle $F_3F_4F_1$. The Fig. 7.3 shows the triangle $F_3F_4F_1$ subjected to the restrictions described above (and thus renamed PQS), and z contours of the corresponding spheroids for fixed $r = 0.295$. The third row in Table 7.1 gives the scaling and rotation necessary to map between $F_3F_4F_1$ and PQS. The point F_5 is relocated by the scaling, rotation and translation from $(5, 4, 3)$ to $(1.52, -0.690, 1.77)$, approximately. This point is denoted F_5'. Since all the points have shifted, the distances between the corners of the triangle and F_5' have changed, too. Consequently, the values in Table 7.2 do not equal the distances in the PQS setup. However, the scaling (which is the only operation that matters in this context) scales the distances equally, and since the scaling is known the (simulated) measurements can be converted to match the PQS setup by division by γ. Note, however, that this division is not necessary in relation to the intersection function presented shortly since the r factor in the spheroid equations also scales the measurements equally. The measurements needed in the present triangle is $w_1 = F_3F_4$, $w_2 = F_1F_4$, and $w_3 = F_1F_3$ from Table 7.2.

In Fig. 7.3 the contours of the three spheroids with $r = 0.295$ are shown for $z = 0$ and $z = 1.77$. Note how the contours all meet in F_5' for the latter choice of z. In the following sections the relation between $\mathbf{w}$, the (x, y, z) coordinate of the intersection, and r is given. For instance, it will be evident that for a given admissible choice of $\mathbf{w}$ (meaning that it comes from a point in 3D) there is a unique vector function $\mathcal{U} : \mathbb{R}^+ \mapsto \mathbb{R}^2 \otimes \mathbb{R}^+$ mapping r to a space curve such that the correct 3D point (the one which corresponds to $\mathbf{w}$) is mapped to exactly when $r = 1/(2\gamma)$.

It is important to note that in a real setup the 'admissible measurements' are known up to a scaling factor, which means that it is not $\mathbf{w}$, but rather $a\mathbf{w}$ for some unknown a that fits the description in the previous paragraph. Thus, knowing γ is not sufficient information to determine the correct point in 3D. The r factor has been introduced for the very purpose of accommodating this particular 'lack of information'.

7.3.5 Intersection for Fixed Focal Points

The intersection function for the restricted setup is presented in this section. The function is derived on the basis of purely geometrical consideration, namely by solving the three spheroid equations simultaneously.

Notation 7.5 (The Positive Orthant)
The positive orthant of $\mathbb{R}^N$ is that subset of $\mathbb{R}^N$ for which all coordinates are ≥ 0. This subset is denoted $\mathbb{R}^{N+}$. For $N = 1$ the notation $\mathbb{R}^+$ is used.

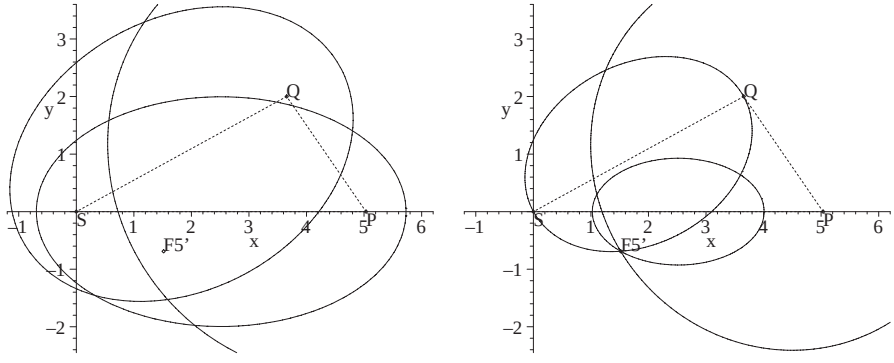


Figure 7.3: The triangle PQS is $F_3F_4F_1$ scaled, rotated, and shifted according to Table 7.1. The left plot shows the contour of the corresponding spheroids for $z = 0$, and the right plot shows the contour of the spheroids for $z = 1.77$. In both plots $r = 0.295$. The values in Table 7.1 and 7.2 have been used.

In the following we will need quantities on the form $w_n - w_k$ several times. The notation w_{nk} will be used as an 'acronym' for this.

Lemma 7.6
The (x, y, z) *solution to the set of equations (7.6), (7.7), and (7.8) for which* $z \geq 0$ *when* $\mathbf{w}$ *corresponds to a point in* (x, y, z) *space is a vector function* $\mathbb{R}^{6+} \mapsto \mathbb{R}^2 \otimes \mathbb{C}$ *given by*

$$\widetilde{\mathcal{U}}^{\triangle}(\mathbf{w}, \mathbf{d}, r) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{d_1} \begin{bmatrix} w_{21}w_3r^2 + d_1^2 \\ (d_1w_2w_{31} + d_2w_3w_{12})r^2 + d_1(1 - d_1d_2 + d_2^2) \\ \sqrt{A^{\triangle}(\mathbf{w}, \mathbf{d})r^4 + B^{\triangle}(\mathbf{w}, \mathbf{d})r^2 + C^{\triangle}(\mathbf{d})} \end{bmatrix}, \quad (7.9)$$

where

$$\begin{aligned} A^{\triangle}(\mathbf{w}, \mathbf{d}) &= -\big[d_2w_{12}w_3 - d_1w_{13}w_2\big]^2 - w_{12}^2w_3^2 \\ B^{\triangle}(\mathbf{w}, \mathbf{d}) &= 2d_1d_2w_3w_{12}(d_1d_2 - d_2^2 - 1) + d_1^2\big(w_{23}^2 + 2d_2w_2w_{13}(d_2 - d_1) + w_1^2\big) \\ C^{\triangle}(\mathbf{d}) &= -d_1^2(d_2^2 + 1)((d_1 - d_2)^2 + 1) \ . \end{aligned}$$

This and some of the following proofs describe fairly simple constructions, where the intermediate formulas are huge expressions (some could fill an entire page). Consequently, they have been left out, since they in this do not serve any important purpose. The following proof is shown with more details in Appendix B.

Proof
A proper scaling of (7.8) followed by a subtraction of (7.8) from (7.7) eliminates z, and a second degree equation in x emerges. (The trigonometric functions resolve nicely). The two solutions to this equation are then inserted in (7.6) and (7.8). Another scaling

followed by a subtraction yields two other second degree polynomials in y, which then gives four candidates for y. Inserting those along with the corresponding x in (7.7) gives a total of eight candidates for z, all on the this $\pm\sqrt{\cdot}$ form. Since we are interested in $z \geq 0$, we are left with four z candidates. The true solution is found by choosing a point in space and three focal points, determine the corresponding $\mathbf{w}$, $\mathbf{d}$, and r, inserting this set of arguments into z. Only one candidate will then yield a real z coordinate. This z and the corresponding x and y are given in (7.9). □

The lemma gives the form of one of the solutions to spheroid equations. There are actually eight solutions (as hinted in the proof) because the equations involves second degree terms of all three variables. The other solutions have similar forms, but unlike the one given above they do not give a $z \in \mathbb{R}^+$ for the coordinate (x, y, z) that matches $\mathbf{w}$ (recall that $\mathbf{w}$ is completely determined by the coordinates of the reflecting object and the locations of the sensors).

It is obvious from (7.9) that the x and y coordinates will be real independently of the choice of $\mathbf{w}$ and $\mathbf{d}$. This is not the case for z, however. Because on the one hand a choice of $\mathbf{w}$ which makes the spheroids too small or too large to intersect cannot give a real z for any r, and on the other hand choosing $\mathbf{w}$ such that it matches a particular point in $\mathbb{R}^2 \otimes \mathbb{R}^+$ must give a real z for some r. Based on this, the previous observations, and a continuity argument we can conclude the following.

Lemma 7.7
The z in $\widetilde{\mathcal{U}}^{\triangle}(\mathbf{w}, \mathbf{d}, r)$ in Lemma 7.6 is real when and only when there exists $t \in \mathbb{R}^+$ such that

(i) $\widetilde{\mathcal{U}}^{\triangle}(t\mathbf{w}, \mathbf{d}, r) \in \mathbb{R}^2 \otimes \mathbb{R}^+$
(ii) *and r belongs to an interval on the form*

$$[\frac{t}{2\gamma} - e_1; \frac{t}{2\gamma} + e_2] \subset \mathbb{R}^+$$

where $e_1, e_2 > 0$.

Note that $\widetilde{\mathcal{U}}^{\triangle}$ was constructed under the assumption that $\gamma = 1$, and the lemma therefore currently applies only in this case. However, later it will be evident that this restriction have no influence on the observations that lead to this lemma, and consequently the lemma also holds in the general case presented in Section 7.3.6.

Based on the lemma it is easy to give a definition of admissible measurements; it is exactly those points $\mathbf{w} \in \mathbb{R}^{3+}$ which corresponds to a point (x, y, z) in $\mathbb{R}^2 \otimes \mathbb{R}^+$, as stated in the lemma.

Definition 7.8
Let P, Q, and S be three points satisfying definition 7.2. The set $\mathcal{F}^{\triangle}_{PQS}$ is defined as the set of vectors $\mathbf{u} \in \mathbb{R}^{3+}$ for which there exist a point $P_0 = (x, y, z) \in \mathbb{R}^2 \otimes \mathbb{R}^+$ and $r > 0$

such that

$$\begin{aligned} ru_1 &= \text{dist}(P_0, P) + \text{dist}(P_0, Q), \\ ru_2 &= \text{dist}(P_0, Q) + \text{dist}(P_0, S), \\ ru_3 &= \text{dist}(P_0, P) + \text{dist}(P_0, S). \end{aligned}$$

The same observations made for intersections of the spheroids can be made for intersection of the spheres. It is therefore relevant to have the following definition.

Definition 7.9
Let P, Q, and S be three points satisfying definition 7.2. The set $\mathcal{F}^{\circ}_{PQS}$ is defined as the set of vectors $\mathbf{u} \in \mathbb{R}^{3+}$ for which there exist a point $P_0 = (x, y, z) \in \mathbb{R}^2 \otimes \mathbb{R}^+$ and $r > 0$ such that

$$\begin{aligned} ru_1 &= 2 \cdot \text{dist}(P_0, P), \\ ru_2 &= 2 \cdot \text{dist}(P_0, Q), \\ ru_3 &= 2 \cdot \text{dist}(P_0, S). \end{aligned}$$

We are now finally ready to define the intersection set for the spheroids.

Theorem 7.10 (The Particular Intersection Function for Triangles)
The three spheroids (7.6), (7.7), and (7.8) intersect iff $r\mathbf{w} \in \mathcal{F}^{\triangle}_{PQS}$. In this case $\widetilde{\mathcal{U}}^{\triangle}(r)$ is mapping $I \subset \mathbb{R}^+$ into $\mathbb{R}^2 \otimes \mathbb{R}^+$, where I is a compact set.

For future reference define $D^{\triangle} = (B^{\triangle})^2 - 4A^{\triangle}C^{\triangle}$. Since it is assumed that there is a sensor at each focal point, there will also be measurements available for the reflected intensity of light emitted from and received at the same point (the same sensor).

Theorem 7.11 (The Particular Intersection Function for Spheres)
The exists an $(x, y, z) \in \mathbb{R}^2 \otimes \mathbb{R}^+$ solution to the following set of spheres equations

$$P: \quad (x - 2d_1)^2 + y^2 + z^2 = v_1^2 r^2 \tag{7.10}$$

$$Q: \quad (x - 2d_2)^2 + (y - 2)^2 + z^2 = v_2^2 r^2 \tag{7.11}$$

$$S: \quad x^2 + y^2 + z^2 = v_3^2 r^2 \tag{7.12}$$

iff $r\mathbf{v} \in \mathcal{F}^{\circ}_{PQS}$, and this solution is given by

$$\widetilde{\mathcal{U}}^{\circ}(\mathbf{v}, \mathbf{d}, r) = \frac{1}{4d_1} \begin{bmatrix} (v_3^2 - v_1^2)r^2 + 4d_1^2 \\ (d_1(v_3^2 - v_2^2) - d_2(v_3^2 - v_1^2))r^2 + 4d_1(1 - d_1 d_2 + d_2^2) \\ \sqrt{A^{\circ}(\mathbf{w}, \mathbf{d})r^4 + B^{\circ}(\mathbf{w}, \mathbf{d})r^2 + C^{\circ}(\mathbf{d})} \end{bmatrix} \tag{7.13}$$

where

$$A^{\circ}(\mathbf{v}, \mathbf{d}) = -\big(d_1(v_2^2 - v_3^2) + d_2(v_3^2 - v_1^2)\big)^2 - (v_1^2 - v_3^2)^2$$

$$B^\circ(\mathbf{v}, \mathbf{d}) = 8d_1\big((d_1 - d_2)(d_2^2 + 1)v_1^2 + d_1(d_2(d_2 - d_1) + 1)v_2^2 + d_2((d_1 - d_2)^2 + 1)v_3^2\big)$$
$$C^\circ(\mathbf{d}) = -16d_1^2(d_2^2 + 1)\big((d_1 - d_2)^2 + 1\big) .$$

In this case $\widetilde{\mathcal{U}}^\circ(r)$ is mapping $I \subset \mathbb{R}^+$ into $\mathbb{R}^2 \otimes \mathbb{R}^+$, where I is a compact set.

For future reference define $D^\circ = (B^\circ)^2 - 4A^\circ C^\circ$.

To illustrate the result of applying $\widetilde{\mathcal{U}}^\triangle(r)$ to an actual case Fig. 7.4 shows three intersection sets. They appear to be well-behaved, and it is demonstrated in the next section that this is actually always the case for intersection sets of spheroids and spheres (under the given assumptions and constraints presented previously in this chapter).

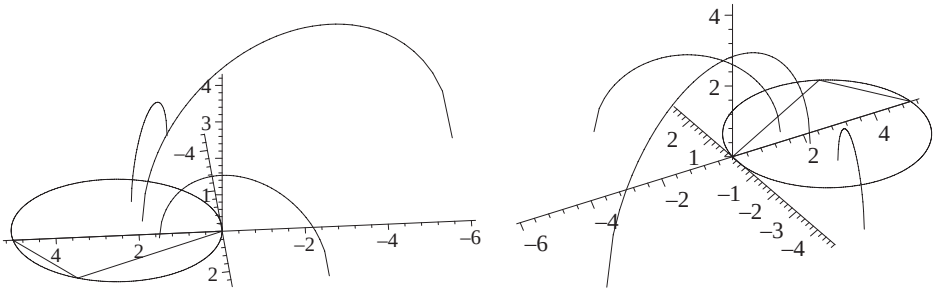


Figure 7.4: Three intersection curves generated by $\widetilde{\mathcal{U}}^\triangle(\mathbf{w}, \mathbf{d}, r)$ with values from Table 7.2 and 7.1. The three curves (clock-wise seen from the center of the circle) have w_2 modified by -1.5, 0, and 1.2. The intervals for r are [0.275; 0.560], [0.236; 0.854], and [0.233; 2.14], respectively. Note that they all include $1/(2\gamma) = 0.295$.

7.3.6 The General Intersection Function

Having introduced the particular intersection functions which are valid only for sensors located at rather restricted locations in the xy plane, we are now ready to relax some of these conditions. This is done by returning to the original definition of sensor locations, that is Definition 7.2. To extend the intersection function presented in the previous section all that is necessary is to perform the inverse of the scaling, rotation, and translation which was applied to the general case in order to restrict it to the particular case of Section 7.3.3.

At the same time the variables γ, θ, d_1, and d_2 are 'hidden' in the general intersection function since they relate, in a sense, to the particular case, whereas the PQS notation is more natural in the general case. A new and more simple definition of the intersecting objects is therefore also given. It is based on the immediately available information, that is the sensor locations, rather than the derived quantities θ, γ, d_1, and d_2.

Definition 7.12 (Prolate Spheriod II)
Let $\mathcal{E}(H, G, a)$ denote the locus of a prolate spheroid constructed by revolving an ellipse with focal points in H and G around the semimajor axis a.

Note that $\mathcal{E}(H, H, a)$ will give a sphere with centre in H and radius a.

This definition allows a simple formulation of the general intersection functions.

Theorem 7.13 (The Intersection Functions)
Let P, Q, and S be three points satisfying definition 7.2. The three spheroids $\mathcal{E}(P, Q, w_1 r)$, $\mathcal{E}(Q, S, w_2 r)$, and $\mathcal{E}(P, S, w_3 r)$ intersect iff $r\mathbf{w} \in \mathcal{F}^{\triangle}_{PQS}$. The intersection point is given by

$$\mathcal{U}^{\triangle}_{PQS}(\mathbf{w}, r) = \gamma \mathbf{G}(-\theta)\widetilde{\mathcal{U}}^{\triangle}(\mathbf{w}, \mathbf{d}, r) + [S_1\ S_2\ 0]^{\top}. \tag{7.14}$$

Equivalently, the three spheres $\mathcal{E}(P, P, v_1 r)$, $\mathcal{E}(Q, Q, v_2 r)$, and $\mathcal{E}(S, S, v_3 r)$ intersect iff $r\mathbf{v} \in \mathcal{F}^{\circ}_{PQS}$. The intersection point is given by

$$\mathcal{U}^{\circ}_{PQS}(\mathbf{v}, r) = \gamma \mathbf{G}(-\theta)\widetilde{\mathcal{U}}^{\circ}(\mathbf{v}, \mathbf{d}, r) + [S_1\ S_2\ 0]^{\top}. \tag{7.15}$$

Note that the functions $\mathcal{U}^{\triangle}_{PQS}(\mathbf{w}, r)$ and $\mathcal{U}^{\circ}_{PQS}(\mathbf{v}, r)$ are undefined when $r\mathbf{w} \notin \mathcal{F}^{\triangle}_{PQS}$ and $r\mathbf{v} \notin \mathcal{F}^{\circ}_{PQS}$, respectively.

It was stated earlier that the intersection set generated by a varying r and fixed $\mathbf{w}$ produced a well-behaved curve. This was also demonstrated for a few examples of $\mathbf{w}$ in Fig. 7.4. The following lemma shows that this is indeed always the case, and, moreover, that this set is always a half circle.

Lemma 7.14
Let P, Q, and S be three points satisfying Definition 7.2, and let $\mathbf{w} \in \mathcal{F}^{\triangle}_{PQS}$. Then $\mathcal{U}^{\triangle}_{PQS}(r)$ equals the intersection of $\mathbb{R}^2 \otimes \mathbb{R}^+$ and a circle with centre in the xy plane. The projection of this circle onto the xy plane is a part of a line which goes through the center of the circumscribed circle to the triangle PQS. This also holds for $\mathcal{U}^{\circ}_{PQS}(r)$ with $\mathbf{v} \in \mathcal{F}^{\circ}_{PQS}$.

The projection on the xy plane of an intersection set is shown in Fig. 7.5.

Proof
The projection of $\widetilde{\mathcal{U}}^{\triangle}$ onto the xy plane is

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{d_1}\begin{bmatrix} w_{21}w_3 \\ d_1 w_2 w_{31} + d_2 w_3 w_{12} \end{bmatrix} r^2 + \begin{bmatrix} d_1 \\ 1 - d_1 d_2 + d_2^2 \end{bmatrix}. \tag{7.16}$$

For any fixed choice of $\mathbf{w}$ and $\mathbf{d}$ (7.16) is a straight line through $[d_1,\ 1 - d_1 d_2 + d_2^2]$. The center of the circumscribed circle is the intersection of the perpendicular bisectors. Two of these are given by

$$QS:\ y = -d_2 x + d_2^2 + 1, \qquad PS:\ x = d_1,$$

and the intersection of these are the point $(d_1,\ 1 - d_1 d_2 + d_2^2)$. Showing that $\widetilde{\mathcal{U}}$ describes a circle is done in two steps. First $\widetilde{\mathcal{U}}$ is rotated around the z axis such that the y coordinate becomes independent of r, then the xz coordinates are shown to described a plane circle.

The angle between the line (7.16) and the x axis is

$$\theta = \arctan\Big(\frac{d_1 w_2 w_{31} + d_2 w_3 w_{12}}{w_{21} w_3}\Big).$$

By applying $\mathbf{G}(\theta)$ to $\widetilde{\mathcal{U}}$, the x and y coordinates becomes two large expressions, where the y coordinate is independent of r. Isolating r in the x coordinate and inserting into the z coordinate yields

$$z^2 = -x^2 + \frac{p_1}{p_3}x + \frac{p_2}{p_3} \qquad \Leftrightarrow \qquad z^2 + (x - \frac{p_1}{2p_3})^2 = \frac{4p_2p_3 - p_1^2}{4p_3^2}, \qquad (7.17)$$

where p_n are multinomials in $\mathbf{d}$ and $\mathbf{w}$ (the expressions are given in Appendix B). Since the properties stated in the lemma are independent of rotation, scaling, and translation it follows that it not only applies to $\widetilde{\mathcal{U}}^{\triangle}$, but also to $\mathcal{U}^{\triangle}$.

The proof for $\mathcal{U}^{\circ}(r)$ is equivalent. □

The existence of two different intersection functions for the same set of sensors might

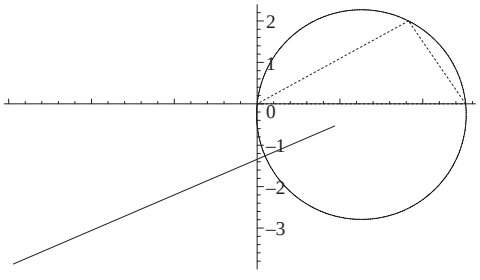


Figure 7.5: The projection onto the xy plane of an intersection set is part of a line which goes through the centre of the circumscribed circle. Values from Table 7.2 are used.

give the impression that the unknown variable r can be determined by finding the point in which the two function intersect (they have to since they both include the point corresponding to **w** and **v**). But as the following lemma shows the two intersection functions provide exactly the same information. Consequently, the r cannot be determined by correlation of the two functions.

Lemma 7.15
Let PQS satisfy Definition 7.2. Let $\mathbf{w} \in \mathcal{F}^{\triangle}_{PQS}$ and $\mathbf{v} \in \mathcal{F}^{\circ}_{PQS}$ correspond to the same point in $\mathbb{R}^2 \otimes \mathbb{R}^+$. Then

(i) $r\mathbf{w} \in \mathcal{F}^{\triangle}_{PQS}, r \in \mathbb{R}^+$ *iff* $r\mathbf{v} \in \mathcal{F}^{\circ}_{PQS}$, *and*

(ii) $\mathcal{U}^{\triangle}_{PQS}(\mathbf{w}, r) = \mathcal{U}^{\circ}_{PQS}(\mathbf{v}, r)$ *for all* r *where* $r\mathbf{w} \in \mathcal{F}^{\triangle}_{PQS}$.

Proof
First *(ii)* is shown by substituting

$$\mathbf{w} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}\mathbf{v}$$

in $\mathcal{U}^{\triangle}_{PQS}(\mathbf{w}, r)$. Then *(i)* follows from Lemma 7.14. □

Since the two intersection functions are equal the notation $\mathcal{U}_{PQS}$ will be used whenever the function expression does not matter (this is usually the case in theory, and usually not the case when using real (noisy) measurements).

Finally, the nice behaviour of the intersection functions inspires a conjecture for another nice property.

Conjecture 7.16
Let any two of w_1, w_2, w_3 *be fixed, and the third varying. Then the locus given by*

$$\left\{\mathcal{U}^{\triangle}_{PQS}(\mathbf{w}, r) \;\middle|\; r\mathbf{w} \in \mathcal{F}^{\triangle}_{PQS}, r \in \mathbb{R}^{+}\right\}$$

equals the intersection of $\mathbb{R}^2 \otimes \mathbb{R}^{+}$ *and a sphere with centre in the* xy *plane.*

The basis for this conjecture is given in Fig. 7.6, where the three loci defined in the conjecture are shown. To further justify it the z contours are shown in Fig. 7.7.

7.3.7 Combining Several Sensors

To find the location of a reflecting object it is not enough to have three sensors and an intersection function (or indeed two intersection functions based on the same set of three sensors, as demonstrated in Lemma 7.15). There is one unknown variable still to be determined. The r in the intersection functions cannot be determined based on reflection information from three sensors. It is therefore necessary to introduce a fourth sensor. This will result in a total of six spheroids, and any combination of three of those will give an intersection function $\mathcal{U}$. In general, for a setup with N sensors located in such a way that any combination of three sensors describes a triangle there is a total of

$$\binom{N}{3} = \frac{N!}{6(N-3)!}$$

spheroid combinations. For any combination of two sensors there is a measurement and for each sensor there is further one measurement (from the sensor to the object and back to the sensor). Consequently, the total number of measurements with N sensors is

$$\binom{N}{2} + N = \frac{N!}{2(N-2)!} + N = \frac{N(N+1)}{2}\,.$$

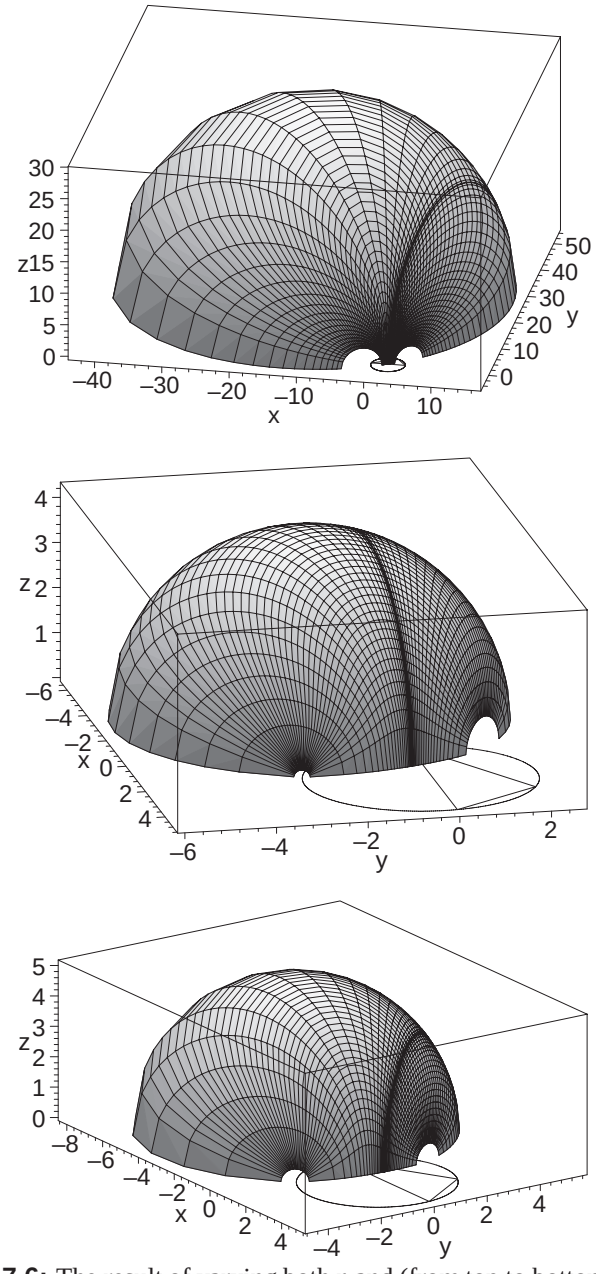


Figure 7.6: The result of varying both r and (from top to bottom) w_1, w_2, or w_3 subject to $r\mathbf{w} \in \mathcal{F}_{PQS}$. All other values are from Table 7.1 and 7.2.

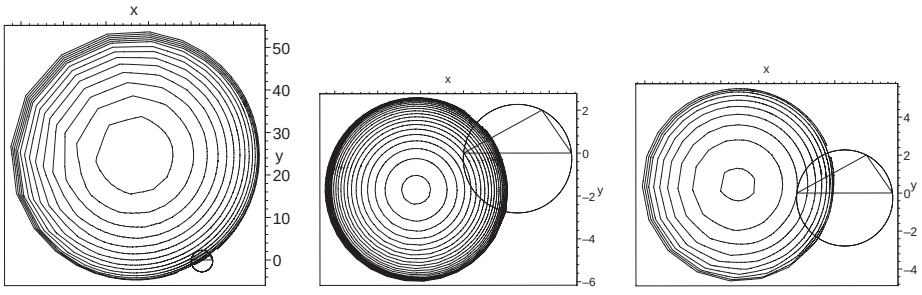


Figure 7.7: The same objects as Fig. 7.6 are shown here with z contours.

Table 7.3: Number of sensor and measurements.

# of sensors (N)	3	4	5	6	7	8	9	10
Spheroid combinations	1	4	10	20	35	56	84	120
# of measurements	6	10	15	21	28	36	45	55

These two number for $N = 3, \ldots, 10$ are given in Table 7.3. It is obvious that four sensors provide plenty of information for determining r. In a real life setup this redundancy can be very useful to reduce the effect of noise. However, it is not immediately clear how to do this.

7.4 Locations of Sensors

Introducing a multiplicity of sensors brings up a new and rather important question. The complete freedom in the locations of the sensors (except no three sensor can be on a line) makes it relevant to ask what the optimal locations are for N sensors. This question relates to real applications, since the task of finding the location of the reflecting object is trivial (once the intersection function is given) in a theoretical setting. Optimality of sensor locations in this context means locating the sensor such as to have the determination of the location of the reflecting object being least sensitive to factors such as measurement noise, hardware degeneration, finite accuracy, rounding errors etc. which inevitably will affect the 'goodness' of the conversion from high dimensional data to three dimensions. Finding the optimal sensor locations thus becomes a matter of combining the influence of each of these factors with the behaviour of the intersection function. Note that the four sensors in Fig. 7.2 are not in any way claimed to be optimally located, they are merely located in what seems to be a nice and close-to-symmetric way.

Although optimality to a great extent depends on a priori unknown factors there are still some theoretical consideration worth doing. In fact, some choices of sensor locations leads to high sensitivity without the possibility of reducing it by either software and hardware solutions. For the purpose of considering theoretical optimality the first step is to determine which mathematical properties of the intersection functions have any influence in this context, and the second step is to determine which model parameters governs these properties.

7.4.1 Optimal Locations

The location of the reflecting object is found as the common point of the intersection curves described in the previous sections. Without noise this point is uniquely defined since all intersection curves coincide at the same point. With noise chances are that no two intersection curves coincide. In the latter case some method is needed to determine which point is 'best' or 'closer' to the right point. This method is necessarily based on not just a single point on each intersection curve, but rather on a interval of the curve, or indeed the whole curve. With two intersection curves a possible solution is to determine the smallest distance between (points on) the two curves, and then let the 'intersection point' be the point which is located half way in between. The question of usefulness of this particular approach is left unanswered at this point. It is easy to come up with variations of this idea, and they all share the need for finding some distance between two (or more) curves. Such an operation is less sensitive when the curves are closer to being perpendicular than parallel. The primary question is therefore (in regards to sensor locations) how to control the intersection curves such as to comply with the desire to have 'mostly perpendicular curves'.

This is to some extent easily answered by Lemma 7.14 which states that any intersection curve projected onto the xy plane is part of a line going through the centre of the circumscribed circle. Thus, having the centres well separated guarantees a not insignificant angular difference between the curves. Moreover, the curves exist in three dimensions, a fact that can cause an increase, but never decrease, in the angular difference.

This raises two new question: 1) What is the optimal location of the centres, and 2) how can the centres be placed in a given pattern? The latter question is relevant since the locations of the centres are completely determined by the location of the sensors.

There are one important observation relating to the first question. Whenever the (x, y) coordinate of the common point of the intersection curves (i.e. the projection of the position of the reflecting object onto the xy plane) lies within the convex hull spanned by the centres of the circumscribed circles (which is a quadrilateral with four sensors) there is a lower limit determined by the 'flatness' of the convex hull to the angles between the intersection curves. No such limit exists outside this convex hull. This lower limit is relatively high when the 'flatness' of the convex hull is small. At least for a four sensor setup this observation is in favor of a large, close to being square, quadrilateral.

But the two questions cannot be finally answered independently, especially not in a real setup which is subject to physical constraints. The dependency between the location of sensors (or more accurately, corners of the triangles) and the centres of the circumscribed circles is by no means linear in behaviour, and consequently small adjustments of the location of a corner might have a significant effect on the location of the centre, and vice versa. Moreover, some pattern of centres cannot be achieved (except in a limit sense).

7.4.2 Sensor Locations and Centres of Circumscribed Circles

It is fairly easy to describe the relation between four sensors and the centres of the circumscribed circles. Since the centre of the circumscribed circle to a triangle is the intersection point of the three (and thus two of) the perpendicular bisectors of the sides, each of the four centres is found as the intersection point of the perpendicular bisectors of two adjacent sides in the (non-intersecting) quadrilateral spanned by the four sensors. This is shown in Fig. 7.8 where the sensor locations are named (x_n, y_n) and the centres are named $(\tilde{x}_n, \tilde{y}_n)$.

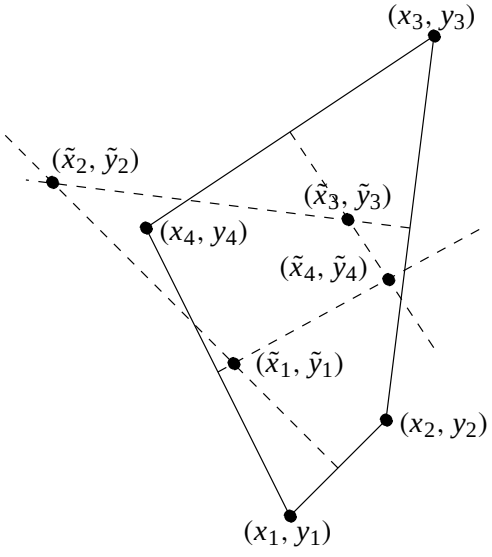


Figure 7.8: Four sensors span a quadrilateral (solid line) and a total of four circumscribed circles can be generated. The four centres each lies on the two perpendicular bisectors (dashed lines) of the sides which are shared by the quadrilateral and the circumscribed triangle.

Two things are immediately noted. Firstly, the quadrilateral spanned by $(\tilde{x}_n, \tilde{y}_n)$ looks like it might be congruent to the quadrilateral spanned by (x_n, y_n). Secondly, the two points $(\tilde{x}_1, \tilde{y}_1)$ and $(\tilde{x}_2, \tilde{y}_2)$ lie on the perpendicular bisector to the side $(x_1, y_1) - (x_2, y_2)$ (and likewise for the three other sides).

The first observation is unfortunately not correct. To each angle v_n in the original quadrilateral there is a corresponding angle ν_n such that $\nu_n = \pi - v_n$. Although this gives a close relation between the two quadrilaterals, it means that they are not in general congruent. They are in special cases, for instance when two opposing sides are parallel. The relation between the two quadrilaterals can be uniquely determined, however. Incidentally, the second observation provides the equations necessary to derive this relation.

Since the line $\tilde{\ell}$ through $(\tilde{x}_1, \tilde{y}_1)$ and $(\tilde{x}_2, \tilde{y}_2)$ is perpendicular to the line ℓ through

(x_1, y_1) and (x_2, y_2) we have

$$\begin{bmatrix} x_1 - x_2 & y_1 - y_2 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 - \tilde{x}_2 \\ \tilde{y}_1 - \tilde{y}_2 \end{bmatrix} = 0 \,, \tag{7.18}$$

and likewise for the three other sides. We also noted that $\tilde{\ell}$ intersects ℓ at the midpoint between (x_1, y_1) and (x_2, y_2). Since the inner product of the normal vector to a line and any point on the line is the same for all points, we also have

$$\begin{bmatrix} \tilde{y}_2 - \tilde{y}_1 & \tilde{x}_1 - \tilde{x}_2 \end{bmatrix} \begin{bmatrix} x_1 + x_2 \\ y_1 + y_2 \end{bmatrix} = 2 \begin{bmatrix} \tilde{y}_2 - \tilde{y}_1 & \tilde{x}_1 - \tilde{x}_2 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \end{bmatrix}, \tag{7.19}$$

and likewise for the three other sides. It is possible to find other equations describing the relations, but the ones presented here have one nice property; they are linear in the unknowns (x_1, y_1) through (x_4, y_4). Expanding the eight equations gives the following equation to be solved

$$\begin{bmatrix}
\tilde{x}_1-\tilde{x}_2 & \tilde{y}_1-\tilde{y}_2 & \tilde{x}_2-\tilde{x}_1 & \tilde{y}_2-\tilde{y}_1 & 0 & 0 & 0 & 0 \\
0 & 0 & \tilde{x}_2-\tilde{x}_3 & \tilde{y}_2-\tilde{y}_3 & \tilde{x}_3-\tilde{x}_2 & \tilde{y}_3-\tilde{y}_2 & 0 & 0 \\
0 & 0 & 0 & 0 & \tilde{x}_3-\tilde{x}_4 & \tilde{y}_3-\tilde{y}_4 & \tilde{x}_4-\tilde{x}_3 & \tilde{y}_4-\tilde{y}_3 \\
\tilde{x}_1-\tilde{x}_4 & \tilde{y}_1-\tilde{y}_4 & 0 & 0 & 0 & 0 & \tilde{x}_4-\tilde{x}_1 & \tilde{y}_4-\tilde{y}_1 \\
\tilde{y}_1-\tilde{y}_2 & \tilde{x}_2-\tilde{x}_1 & \tilde{y}_1-\tilde{y}_2 & \tilde{x}_2-\tilde{x}_1 & 0 & 0 & 0 & 0 \\
0 & 0 & \tilde{y}_2-\tilde{y}_3 & \tilde{x}_3-\tilde{x}_2 & \tilde{y}_2-\tilde{y}_3 & \tilde{x}_3-\tilde{x}_2 & 0 & 0 \\
0 & 0 & 0 & 0 & \tilde{y}_3-\tilde{y}_4 & \tilde{x}_4-\tilde{x}_3 & \tilde{y}_3-\tilde{y}_4 & \tilde{x}_4-\tilde{x}_3 \\
\tilde{y}_4-\tilde{y}_1 & \tilde{x}_1-\tilde{x}_4 & 0 & 0 & 0 & 0 & \tilde{y}_4-\tilde{y}_1 & \tilde{x}_1-\tilde{x}_4
\end{bmatrix}
\begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{bmatrix}$$

$$= 2 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \tilde{y}_1\tilde{x}_2 - \tilde{x}_1\tilde{y}_2 \\ \tilde{y}_2\tilde{x}_3 - \tilde{x}_2\tilde{y}_3 \\ \tilde{y}_3\tilde{x}_4 - \tilde{x}_3\tilde{y}_4 \\ \tilde{y}_4\tilde{x}_1 - \tilde{x}_4\tilde{y}_1 \end{bmatrix}. \tag{7.20}$$

Note that the first row corresponds to (7.18), while the fifth row corresponds to (7.19). Solving this equation means inverting the square matrix. Fortunately, the matrix has full rank in most cases. Examples of degenerate cases are when the two center points coinciding and when the four centres span a rectangle (in both cases the rank is 6). Assuming that the the matrix is not degenerate the solution to (7.20) is provided directly by inverting the matrix. Unfortunately, the determinant is a huge expression, and consequently, the solutions are not very nice. For instance $x_1 = N/D$ where ($\tilde{x}$ and $\tilde{y}$ are written as x and y in the following two expressions)

$$N = \; y_3^2x_4^2y_1 - y_1^2x_4^2y_3 - x_1^2y_2^2y_3 + x_4^2y_2^2y_3 - x_4^2y_3^2y_2 - y_1^2x_4y_3x_3 - x_2^2y_1^2y_3$$

$$
\begin{aligned}
&- y_2^2y_3x_3x_4 - x_2y_1^2x_4y_4 - x_1^2x_2x_4y_4 - x_1y_2^2x_4y_4 + x_1x_2^2x_4y_4 + 2x_2y_2y_1x_4y_4\\
&- 2y_4x_1x_2y_2y_1 + y_2^2x_1^2y_4 + x_2x_1y_4^2y_2 - x_2^2y_1y_4^2 + x_2x_1y_1y_4^2 + x_2^2y_1^2y_4\\
&- y_2x_1^2y_4^2 + x_1^2y_2y_3^2 + 2y_3y_2x_1x_4y_4 - 2y_3x_2x_1y_4^2 + y_3x_2^2y_4^2 - 2y_3x_2x_4y_2y_4\\
&+ y_3x_1^2y_4^2 - y_3^2x_1y_4x_4 + 2y_3^2x_1x_2y_4 + 2y_1y_3x_3x_4y_2 + y_3^2x_2x_4y_4 - y_3^2x_1^2y_4\\
&- y_3^2x_2^2y_4 - 2x_1y_2y_3y_1x_4 + x_1y_1y_2^2x_4 - x_1x_3y_1y_2^2 + x_1y_2^2y_3x_3 - x_1x_4^2y_1x_3\\
&- x_1x_2y_2y_3^2 + x_1x_4^2y_3x_3 + x_1x_4y_1x_3^2 - x_3^2x_1y_4x_4 + 2x_1x_2y_2y_3y_1 + x_1x_2x_4^2y_1\\
&- x_1x_2x_4^2y_2 - x_1x_2x_3^2y_1 + x_1x_2x_3^2y_2 - x_1x_2y_1y_3^2 + x_1^2x_2x_4y_2 - x_1^2x_4y_3x_3\\
&- x_1x_2^2y_3x_3 + x_1^2x_2y_3x_3 - x_1^2x_2y_2x_3 + x_1x_2^2x_3y_1 - x_1x_2^2y_1x_4 + x_3x_1y_4^2y_3\\
&- x_3x_2y_2y_4^2 - x_3x_4x_2^2y_4 - x_3x_2y_4^2y_3 + x_2y_1^2y_3x_3 + x_2x_4^2x_3y_2 - 2x_2y_1y_3^2x_4\\
&+ 2x_3y_4x_1y_1y_2 + 2x_3y_1x_2y_4^2 - x_3x_1y_1y_4^2 + x_3x_1^2x_4y_4 - 2x_3y_1^2x_2y_4 - y_1x_4^2y_2^2\\
&+ x_3^2y_1y_2^2 + x_1y_1y_3^2x_4 + x_4y_2^2x_3y_4 - 2y_2x_3y_1x_4y_4 - x_3^2y_4y_2^2 + x_3^2y_4^2y_2\\
&- y_2x_3^2y_1^2 - y_1x_3^2y_4^2 + y_1^2y_4x_3^2 + x_4^2y_2y_1^2 + x_4y_4y_1^2x_3 + x_3^2x_2x_4y_4 + x_2x_3y_2y_1^2\\
&- 2x_2y_2y_3y_1x_3 + x_2^2y_1y_3^2 - x_2x_4x_3^2y_2 + x_2^2x_4y_3x_3 + 2x_2y_1^2y_3x_4 - x_2y_2y_1^2x_4\\
&+ x_2y_2y_3^2x_4 - x_2x_4^2y_3x_3 + 2x_3y_3x_2y_2y_4 - 2x_3y_3x_1y_4y_2
\end{aligned}
$$

$$
\begin{aligned}
D = \;& y_4x_3x_1^2 - y_4x_3^2x_1 + x_2y_1^2y_3 + y_2^2y_3x_4 + x_4^2x_3y_2 - x_4x_3^2y_2 + x_2^2y_3x_4 - x_2x_4^2y_3\\
&- y_1^2x_2y_4 + y_1^2x_3y_4 - y_1x_3y_4^2 + y_1x_2y_4^2 + x_1x_2^2y_4 - x_1^2x_2y_4 + x_1^2x_2y_3 + y_2y_1^2x_4\\
&- x_1^2y_2x_3 + x_1^2x_4y_2 - x_1^2y_3x_4 - x_1y_3x_2^2 + x_1x_3^2y_2 + x_1y_2y_3^2 - x_1x_4^2y_2 - y_1^2y_3x_4\\
&+ x_1x_4^2y_3 - x_4^2y_1x_3 - x_1y_2^2y_3 + x_4y_1x_3^2 - y_4x_3y_2^2 + x_2y_4y_3^2 - x_2y_4^2y_3 + x_2^2x_3y_1\\
&- y_2y_3^2x_4 + x_3^2x_2y_4 - x_3x_2^2y_4 - y_1y_2^2x_4 - x_3y_2y_1^2 + x_3y_1y_2^2 + y_1y_3^2x_4 - x_2^2y_1x_4\\
&+ x_1y_2^2y_4 - x_1y_2y_4^2 - x_1y_3^2y_4 + x_1y_4^2y_3 + y_2x_3y_4^2 + x_2x_4^2y_1 - x_2y_1y_3^2 - x_2x_3^2y_1
\end{aligned}
$$

This and the seven other similar expression does not by themselves provide much knowledge on the relation between the location of sensors and the centres of the circumscribed circles. But they do provide easy means for numerical experiments regarding sensor and centre locations.

7.4.3 Examples of Sensor Locations

An important conclusion of the results presented in the previous section is that locating the sensors in a rectangle (or close to a rectangle) is a bad idea in respect to robustness. The sensor locations in Fig. 7.2 are no exception, as Fig. 7.9 shows. Here the four centres of the circumscribed circles are shown along with the intersection curves for the point $(4, 3.5, 2)$. It is immediately evident from the figure that determining the position of the reflecting object is very sensitive to variations in the intersection curves because they are

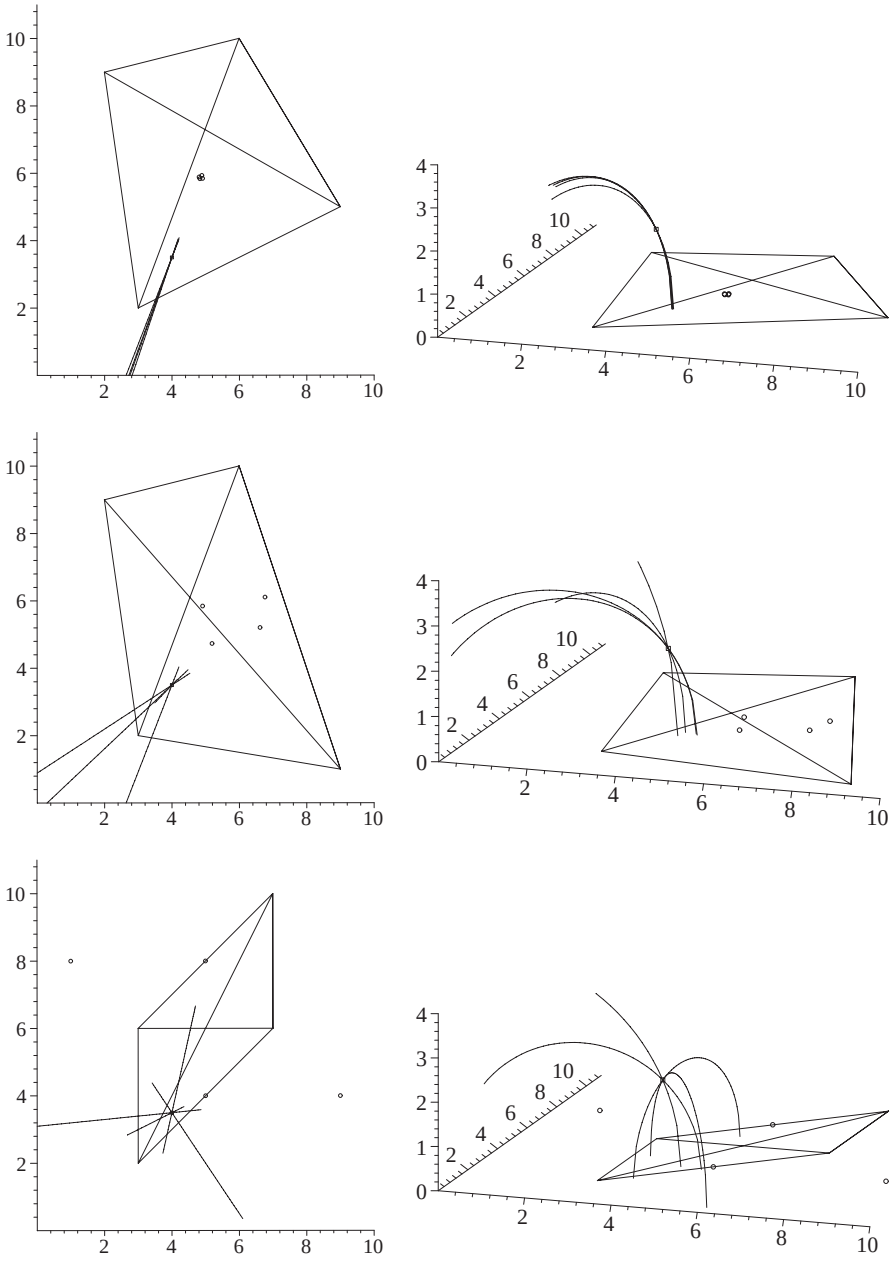


Figure 7.9: Three different locations of sensor. The solid lines show the quadrilateral and triangles spanned by the sensors, while the four small circles shown the centres of the circumscribed circles. The left column shows the projection onto the xy plane.

'almost parallel'. Moving one sensor to another location (here F_2 are moved from $(9, 5)$ to $(9, 1)$) improves the robustness even though the quadrilateral spanned in the latter case seems to be just as close to a rectangle as the quadrilateral in the former case.

It is important to note that the sensitivity is high in the first example independently of the location of the reflecting object since the centres of the circumscribed circle almost coincide. In the second example the sensitivity is reduced inside the quadrilateral spanned by the centres because the directions to the four centres are more different for points in this region compared to points outside this region, as described previously in this section. It is therefore an important observation that the centres now span a significantly larger quadrilateral. Choosing a completely different set of locations, see the third examples in Fig. 7.9, can give a much high robustness since the quadrilateral spanned by the centres is significantly larger than in the second examples.

7.5 Assumptions Revisited

There are a number of differences between the presented model and reality. The most obvious and important ones were presented in Section 7.2 in the beginning of this chapter. They are still valid and they do raise the question on the usability of the model. Being clearly inaccurate the model does not provide the final solution to the mapping from high dimensional data to 3D position, and the usability is therefore more of a qualitatively kind rather than quantitatively. This section briefly discusses the importance of the model inaccuracy, possible ways of handling this, and what effect they have on the solution given by the model.

7.5.1 Emitter and Receiver Characteristics

The choice of ellipses (and thus spheroids) to model the reflection map in Section 7.3 were primarily based on two assumptions. The first was that the emitters and receivers have uniform directional characteristics, the second was that the reflecting object has an ability to reflect light in a certain manner. The former assumption is discussed here, while the second assumption is discussed in Section 7.5.2.

In order to give a qualitatively description of the significance of the directional characteristics of the emitters and receivers Fig. 7.10 shows the contours of the reflection maps under the assumption of uniform and angularly varying directional characteristics (in the first two rows as a cosine, in the third row with a modeled characteristic). The emitter and receiver are located in $(0, -1)$ and $(0, 1)$, respectively, and $\mathbf{e}$ and $\mathbf{r}$ are the vectors from any point in the 2D map to the emitter and receiver, respectively.

The assumption made in this chapter gives the non-diffuse uniform directional reflection map (top left). The contours in this graph are exactly ellipses with focal points in $(0, 1)$ and $(0, -1)$, as was argued in the beginning of this chapter. This also follows from the reflection function shown above the graph. Introducing the angularly varying directional characteristics for emitter and receiver (a cosine in both cases) produces a some-

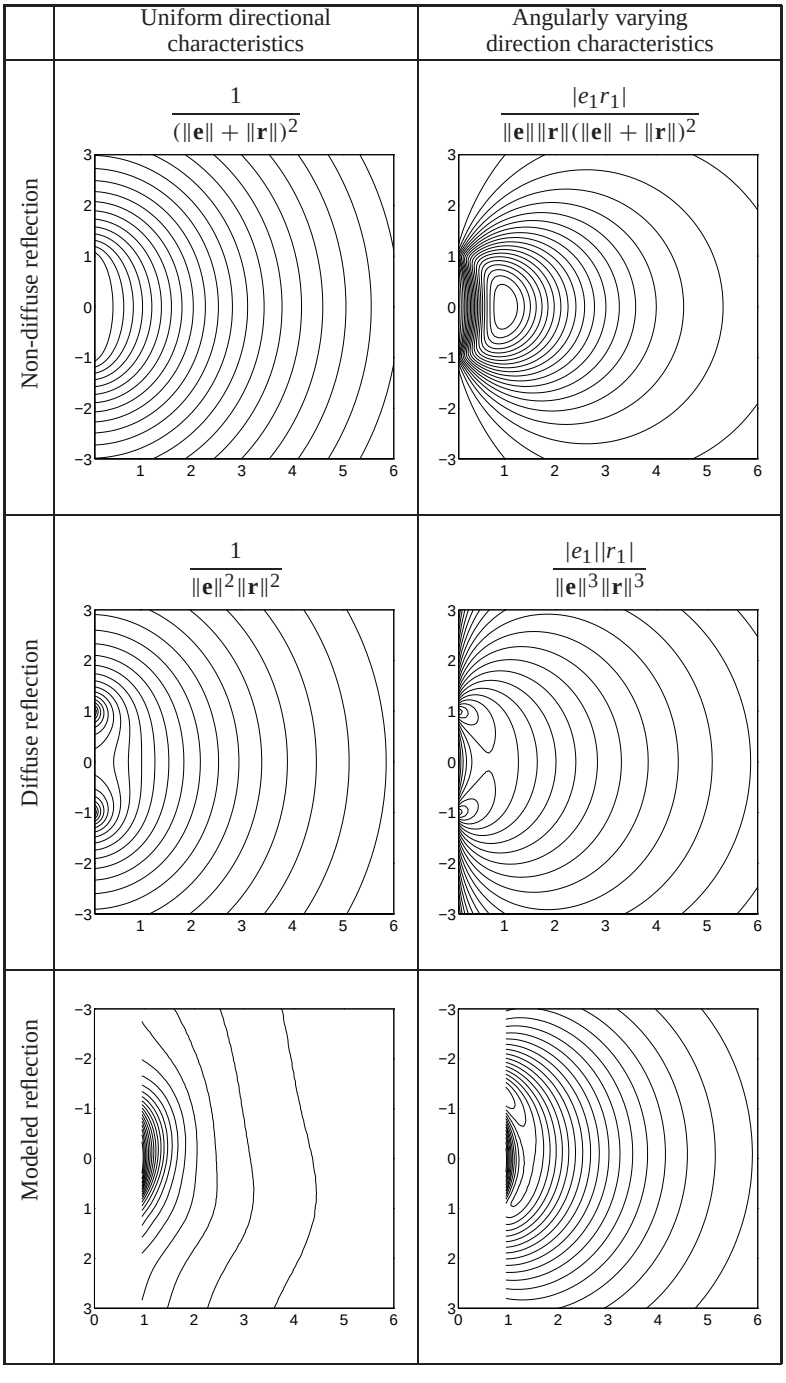


Figure 7.10: Contours for simulated reflection maps under various assumptions (the lower right contour plot is very close to measurement data).

what different contour graph (top right). The difference from the uniform characteristic is apparent, especially in the vicinity of emitter and receiver.

Introducing the diffuse reflection in its generic form actually does not change the contour lines very much except close to the emitter and receiver. Again there is a different between uniform and cosine directional characteristic.

Using a modeled reflection, see Chapter 8, the contour lines become asymmetric and the directional characteristic suddenly seems to have a major influence on the reflection map. And nonetheless the modeled reflection map with cosine characteristics is in general reasonably close to the simple formed used for constructing the geometrical solution in this chapter. Although the difference are too big to be neglected it seems fair to assume that the qualitative conclusions drawn in this chapter does to some extent apply to real setups, too.

7.5.2 Reflecting Object

In the introduction to this chapter it was mentioned that the assumptions for this model are rather unrealistic. Especially the assumptions on the reflecting object seemed far-fetched. The object should reflect the light such that the isocandela sets are prolate spheroids. This is achieved by an intensity function on the form (7.1). This calls for the reflection to be without any scattering and in the direction of the receiver. The former property is easily achieved, whereas the second posses a problem. As it was hinted in the beginning of the chapter it is not possible to have reflection in different directions in the same point. A very small sphere

comes close, but does not obtain the exact property except in a limiting sense. Using a very small sphere might therefore be a good idea, at least in a theoretical setting where the emitter is considered a point source. In a real setting where the emitter has a finite spatial extension the curvature of the reflecting object determines the amount of light reflected in any given direction, in particular towards the receiver, i.e. the smaller the sphere, the smaller the reflection of the emitter looks from the receiver's point of view. Enlarging the sphere does reduce this problem, but at the same time the object becomes more distant from the property of reflecting in different directions from the same point in space.

Perhaps a satisfactory object, i.e. a good trade-off between the opposing desired properties, can constructed in the following way. The object is constructed using a convex polyhedron with a sufficient high number of faces and with (almost) equal angles between them. This could be for instance a trapezoidal or pentagonal icositetrahedron (24 faced polyhedron). The faces need to have the property that incoming light is scatter slightly such that in the case where a face of the polyhedron is almost, but no quite tangent to a spheroid, there is still some light reflected onto the receiver. More precisely, the reflection characteristic of the surface should be such that an incoming ray of light perpendicular to the surface is reflected such that the outgoing rays of light covers a range from $[-\theta;\theta]$, where angle 0 is perpendicular to the surface and where θ is the angle between normal vectors to two adjacent faces on the polyhedron. The principle is shown in Fig. 7.12. As

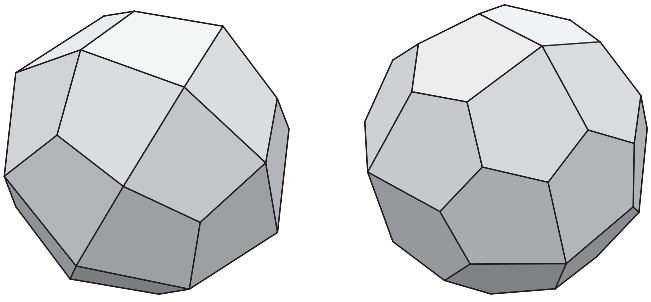


Figure 7.11: A trapezoidal (left) and a pentagonal (right) icositetrahedron.

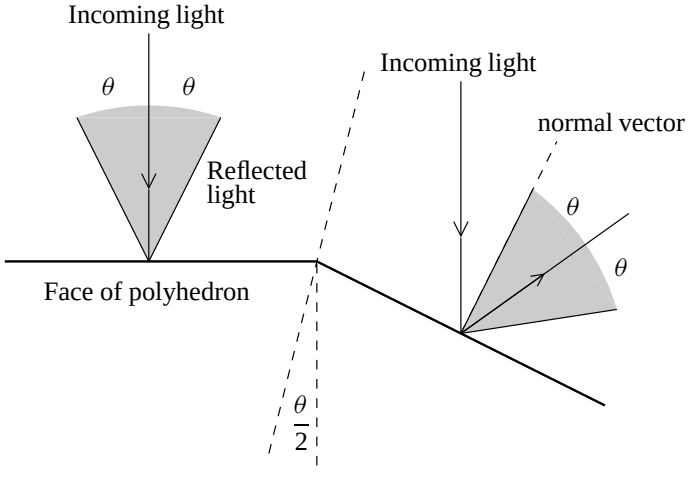


Figure 7.12: The principle for an alternative reflecting object.

this figure demonstrates the object will have the property that any two faces will reflect the incoming light in a way that forms 'disjoint' cones of reflected light. Each cone represents an angle equal to the angle between normal vectors of adjacent faces. This will enable the object to reflect light in all direction (like a sphere) without the incoming light being reflected in any direction by more than one face. Such an object would indeed be tangent to several different spheroids at almost the same point (provided that the object is small), while at the same time taking into account the finite size of the emitter. The downside is the introduction of some degree of scattering. This is kept at a minimum by the many-faced polyhedron, however.

One question remains; how to produce a surface with the desire property. This could be done by a structure like the one shown in Fig. 7.13. The surface structure is shown in a side view, i.e. the same view as in Fig. 7.12. Seen from the top the structure consists of concentric annuli, each with the outer edge higher than the inner edge (except for the center circle, which is flat).

Figure 7.13: A suggestion for the structure of surface with a slight scattering property.

It is not known how well this construction will approximate the fictitious object which is the basis of the spheroidal-based model presented in this chapter.

7.6 Conclusion

A geometrical model of the mapping of CGMs to spatial position has been presented in this chapter. A series of assumptions were imposed to reduce the complexity of the model, and a number of parameters were included to provide a flexible model. The result is a mapping which in some respects is quite useful for determining properties of multiple emitter/receiver setups, but in other respects is inaccurate to an extent which rules out the immediate use in real applications. Despite the fact that the model is quite restricted by the assumptions the model is still rather complex. This does not rule out implementation in signal processing hardware, though.

The result of modeling the multiple emitter/receiver setup is a set of equations which directly maps CGMs to a coordinate in $\mathbb{R}^3$. The geometrical derivation ensures the analytical correctness of the equations, but does not guarantee numerical stability. On the contrary, the equations includes polynomial forms which typically have a inherent instability problem. The author has investigated a method for stabilizing the mapping, but no conclusion has been reached so far, and the results so far are therefore not reported in this thesis.

The question of the optimal locations of the emitters and receivers was also discussed. An analysis of the mapping equations revealed that certain sensor positions were a priori

bad in the sense that the mapping becomes sensitive to disturbances independently of any stabilizing methods applied to the data (such as filtering). In fact, it was demonstrated that placing emitter and receivers at the corners of a square increased the sensitivity to infinite as a singularity in the mapping occurs in that situation. It was also discovered that a best case scenario is application specific as the optimal placement depends on the space available for arranging the emitters and receivers.

7.6.1 Recognition of Objects

An interesting alternative to determining the position of a object given its physical structure is to determine physical structure given its position. That is, recognizing an object in a fixed position. An application could be to determine orientation of gadgets on a conveyor belt in order to allow a robot arm to grab the gadgets (for painting, packaging, etc.). The procedure is in many respects the same as presented here, i.e. acquiring reflectivity information about the object and map it to form or orientation, or use it for classification. The latter is relatively easy if a priori training of the sensor system is possible. The second setup presented in Chapter 5 is originally designed object recognition. It employs three emitters and three receivers, and is currently trained to recognize thirteen different objects. One of them is the LEGO model shown in the fixed position in Fig. 5.5 on page 100. This functionality has not been discussed in this thesis, partly to limit the extent of the thesis, partly because another Ph.D. study focusing on exactly this functionality has just been started to continue the work initiated here.

8 Modeling Reflection Maps

The need for a model of the reflection map in an ‘emitter, receiver, reflecting object’ setup became clear in the previous chapter. In the attempt to determine the position of an object in three dimensions the need for finding intersections of isocandela curves emerged. A series of assumptions lead to prolate spheroids as a model for such curves, but it was also argued that the assumptions were perhaps not very realistic. Consequently, there is still a need to determine how such isocandela curves behave in reality. For the purpose of determining spatial position of an object there is also a need for modeling the curves, and if possible, to parameterize the curves to ease the implementation of the intersection idea presented in the previous chapter.

In this chapter a model involving the emitter and receiver characteristics as well as the reflection characteristics of the object is presented. While the emitter and receiver are modeled according to real physical specifications, the reflecting object is a considered to be completely round, i.e. a ball. This reduces the model considerably as the orientation of the object is then of no concern. The model is constructed in two dimensions in order to reduce the geometrical and the numerical complexity. Obviously, this might turn out to be insufficient since the real setup is, of course, in three dimensions. All references to the round object is therefore ‘circle’, although this conflicts with a rigid interpretation of properties such as the surface of the object. However, there are no mathematical difficulties in having a ‘surface’ in two dimensions.

The first section presents the setup and the individual components. Then, in Section 8.2, the model in the form of an integral equation is developed by a series of geometrical observations. The real reflection intensity map is acquired by measuring on a real setup. This and the resulting measurements are presented in Section 8.3 which also introduces one of two methods for evaluating the model. Then in Section 8.4 the integral equation is rewritten to an inverse problem in order access the robustness of the model. Getting a useful solution requires regularization which is discussed in Section 8.5. Finally, Section 8.6 gives a brief conclusion of the chapter.

8.1 Components in the Model

The entire setup consists of three separate components, namely the emitter, the receiver, and the reflecting object. Each component has a geometrical description which is presented in the following subsections. Since the movable object is reflecting the light it is necessary to include a surface reflection model which is introduced in Section 8.1.3. Finally, a brief introduction to the structure of the model is given in Section 8.1.4. The derivation of the model takes place in Section 8.2

8.1.1 Emitter and Receiver

The emitter is capable of emitting light in many directions with varying intensity. Typically, the data sheet specification of the directional characteristic of an LED is a Gaussian-like function, and to indicate the width of this the half angle is often specified, i.e. the angle between the maximum intensity direction and the half of maximum intensity direction. In many cases, e.g. low-cost LEDs, the true directional characteristic can be significantly different from the data sheet specification. At any rate the directional characteristic of the emitter must be geometrically described to become a part of the reflection map model.

In contrast to the emitter the directional characteristic of the receiver is straight forward. Since it consist of a flat piece of semiconducting material the only quantity (concerning the incoming light) affecting the current generated is the amount of light arriving at the receiver. For parallel light rays this is equal to $\cos\theta$ times the intensity of the light, where θ is the angle deviation between the normal to the receiver and the incoming light.

In the model the emitter is considered infinitely small while the receiver is assumed to have a (small) spatial extension.

8.1.2 Reflecting Object

The emitted light signal can be transferred in four difference ways from to the receiver.

By reflection from small, nearby objects This is (supposed to be) the primary sources of light from the receivers point of view, since the intensity of this light is the basis for determining the channel gain.

By reflection from large, distant objects This could be bright clothing, windows, walls etc. Although the intensity of the emitted light is typically very low large areas such as a wall can still contribute significantly to the amount of reflected light.

Directly from emitter to receiver This could be through the air, through some conductor like glass or plastic, or by reflection from components close to the emitter or receiver (like electrical components and plastic fittings for holding emitter and/or receiver).

Through the wiring and cross talk Since the emitters and receiver often share the power source there is a electrical connection between them. Moreover, to keep the cost low

there is only a sparse screening of the circuits, making cross talk a possible contributor to the received signal.

The undesired contributions are all more or less difficult to reduce, and almost impossible to eliminated completely. Therefore it might be necessary to include some of them in the model. In the case of the measurements presented in Section 8.3.1 precautions have been taken to reduce the impact of 'false' reflection.

To determine the characteristics of the setup (or how well the model fits the true setup) it is imperative to use a known object. To ease the modeling a sphere (a circle in 2D) is chosen. This also eliminates any discussion on the effect of rotating the object.

8.1.3 Surface Reflection Model

The next concern is how the light is reflected from the surface of the circle. For other surfaces than a perfect mirror, the answer is not particularly simple. There is, however, a fairly simple way to approximate the reflection. The idea is as follows: Since light in reality is a vast number of photons, it is instructive to consider the behaviour of just one photon hitting the surface. It seems fair that this one photon is either absorbed or reflected in some direction (which is determined by the properties of the surface), and that the occurrence of both events and the direction is controlled by probability. If this probability is known (e.g. determined by measurements of that surface), it is possible to calculate, for a given number of photons, how many are absorbed and how many are reflected in any given direction. This situation can be approximated by a number of flat, not completely reflecting mirrors whose angles with respect to the surface is determined by the above-mentioned probability. The advantage of this mirror model is that it is easy to predict how light is reflected when nothing but the angle of incidence is known. Let $m(\rho)$, $\rho \in [-\frac{\pi}{2}; \frac{\pi}{2}]$, be the probability density function (p.d.f.) of the angles of the mirrors, where ρ is the angle with respect to the (flat) surface, and let $\kappa \in [0; 1]$ be the reflection coefficient for the (imperfect) mirrors. If for instance the angle of incidence for the light is φ the amount of light reflected at the angles $-\varphi$ must be equal to κ times the probability that the mirrors have angle 0. Generally if the angle of incidence is φ then the amount of light reflected in a given angle interval ρ_1 to ρ_2 must be

$$\kappa \int_{\rho_1}^{\rho_2} m(\rho + \varphi) d\rho \ . \tag{8.1}$$

In Fig. 8.1 the two angles ρ and φ are shown. The p.d.f. for the angle of mirrors could be a normal distribution (Fig. 8.2 left), and the corresponding p.d.f. for the reflected light, when the angle of incidence is φ, is just the normal distribution translated (Fig. 8.2 right).

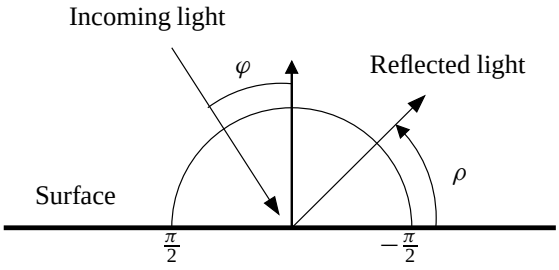


Figure 8.1: The angle of the incoming light is denoted φ and the angle of reflection is denoted ρ. The amount of light reflected in a given direction is determined by the p.d.f. $m(\rho)$.

8.1.4 Model Idea

After considering several approaches for modeling this setup the following seemed most appropriate. The idea is to integrate with respect to the angle of emitted light. Since the emitter covers a half circle, the integration interval is of length π. For convenience the interval $[-\pi/2; \pi/2]$ is chosen. The integrand is then a function which maps angle of emitted light into an intensity at the receiver. This integrand vanishes outside the angle interval corresponding to the extend of the circle. But some intervals on the circle, although receiving light, do not reflect light onto the receiver, since they are 'too far around the circle' to be seen by the receiver. This is exemplified by interval 1 on Fig. 8.3. In the corresponding integration interval the integrand should also vanish. The interval $[\theta_1; \theta_2]$ in which the integrand does not vanish is determined by geometrical observations in Section 8.2.1, and it is denoted emitter integration interval. In Fig. 8.3 the setup is shown with the two angles θ_1 and θ_2. The two radius lines show what part of the circle is within the integration interval. To make the model a little more simple the partly visible interval, no. 2 in the figure, is considered not visible from the receiver. This introduces only a very small error as the spatial extension of the receiver is much smaller than the distance from the circle to the receiver.

For any angle θ the reflection is modeled as described in Section 8.1.3, and it is thereby determined how much light is reflected in the direction of the receiver. This procedure will include the directional characteristic of the receiver (which is described in Section 8.1.1 about the receiver).

8.2 Integral Equation Model

From the descriptions in the previous section it is clear that two integrals are needed. One for the emission of light and one for the reflected light. Hence two integration intervals must be determined. This is done in the following two subsections, starting with the emitter integration interval in Section 8.2.1 followed by the reflector integration interval

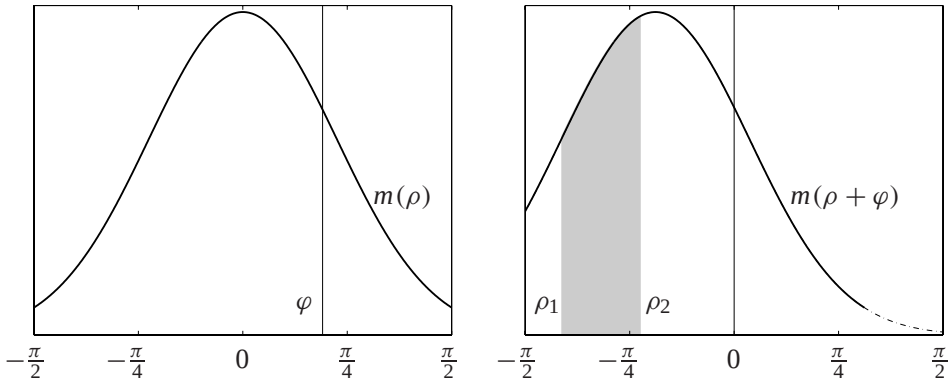


Figure 8.2: To the left is an example of a p.d.f. $m(\rho)$ modeling the surface reflection, where the angle of incidence φ is just below $\pi/4$. To the right is the reflected intensity as a function of angle of reflection. This is just the p.d.f. m translated $-\varphi$. An example of an integration interval $[\rho_1; \rho_2]$ is also shown.

in Section 8.2.2. Then the integral equation is presented in Section 8.2.3. Finally, some examples of using the model is given. The basic components in the model is shown in Fig. 8.4. The two integration variables are θ and ρ for the emitted light and the reflected light, respectively. The center of the reflecting circle is (C_x, C_y) and it has radius R.

8.2.1 Emitter Integration Interval

Since most of the initially known positions (of emitter, receiver, and reflector) are in Cartesian coordinates the angles θ_1 and θ_2 are derived via the slopes of lines through the emitter. The slopes are mapped to angles by arctan.

The first slope $\alpha_1 = \tan\theta_1$ can be found by solving an equation in which the radius R of the circle equals the distance from center of the circle to lines through $(0, E)$. Since $\|\mathbf{a} \times \mathbf{b}\| = \|\mathbf{a}\|\|\mathbf{b}\| \sin v$, where v is the angle between $\mathbf{a}$ and $\mathbf{b}$, this can be accomplished with

$$\frac{\left\| \begin{bmatrix} 1 \\ \alpha_1 \\ 0 \end{bmatrix} \times \begin{bmatrix} C_x \\ C_y - E \\ 0 \end{bmatrix} \right\|}{\sqrt{1+\alpha_1^2}} = R \quad \Leftrightarrow \quad \left(C_y - E - C_x\alpha_1\right)^2 = R^2 + R^2\alpha_1^2 \tag{8.2}$$

Rewriting this yields

$$({C_x}^2 - R^2)\alpha_1^2 + 2C_x(E - C_y)\alpha_1 + (E - C_y)^2 - R^2 = 0.$$

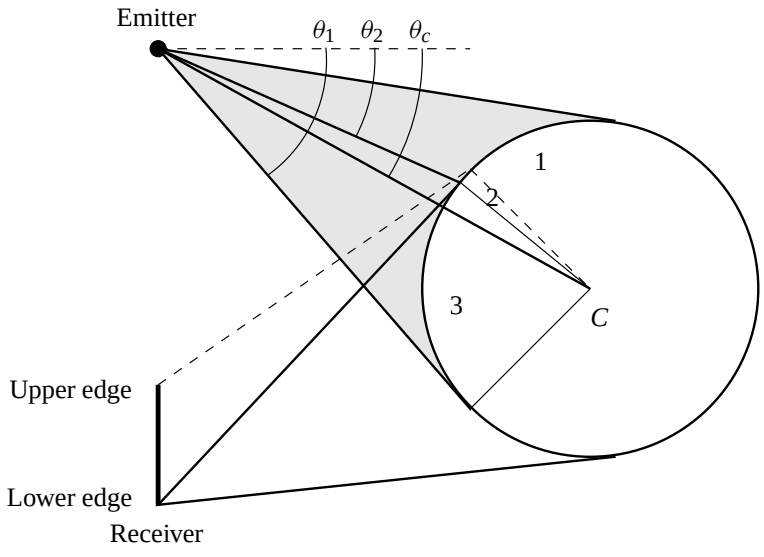


Figure 8.3: The position of the emitter, receiver, and reflecting circle. The integration interval is θ_1 to θ_2. Note that while the emitter is considered infinitely small the receiver has finite length. The three enumerated intervals are all illuminated and (1) invisible, (2) partly visible, (3) visible from receiver.

Since the solution of interest is 'below' the circle (see Fig. 8.3), the desired slope is

$$\alpha_1 = \min \frac{C_x(C_y - E) \pm \sqrt{{C_x}^2(E - C_y)^2 - ({C_x}^2 - R^2)\big((E - C_y)^2 - R^2\big)}}{{C_x}^2 - R^2},$$

which simplifies to

$$\alpha_1 = \frac{C_x(E - C_y) + R\sqrt{{C_x}^2 - R^2 + (C_y - E)^2}}{R^2 - {C_x}^2},$$

because $C_x > R > 0$. The second slope is determined by one of the two points on the circle where the tangent line goes through the lowermost edge of the receiver, that is $(0, R_l)$. (Note that this disregards a small interval on the circle from which only part of the receiver is visible. This is exemplified by interval 2 on Fig. 8.3.) This point on the circle is found by first determining the slope β of the tangent line by an equation like (8.2), then constructing a vector with slope $-1/\beta$ (normal to line) and length R, and finally adding this vector to (C_x, C_y). In formulas the process looks like the following. First the slope is found to be

$$\beta = \frac{C_x(R_l - C_y) - R\sqrt{{C_x}^2 - R^2 + (C_y - R_l)^2}}{R^2 - {C_x}^2}.$$

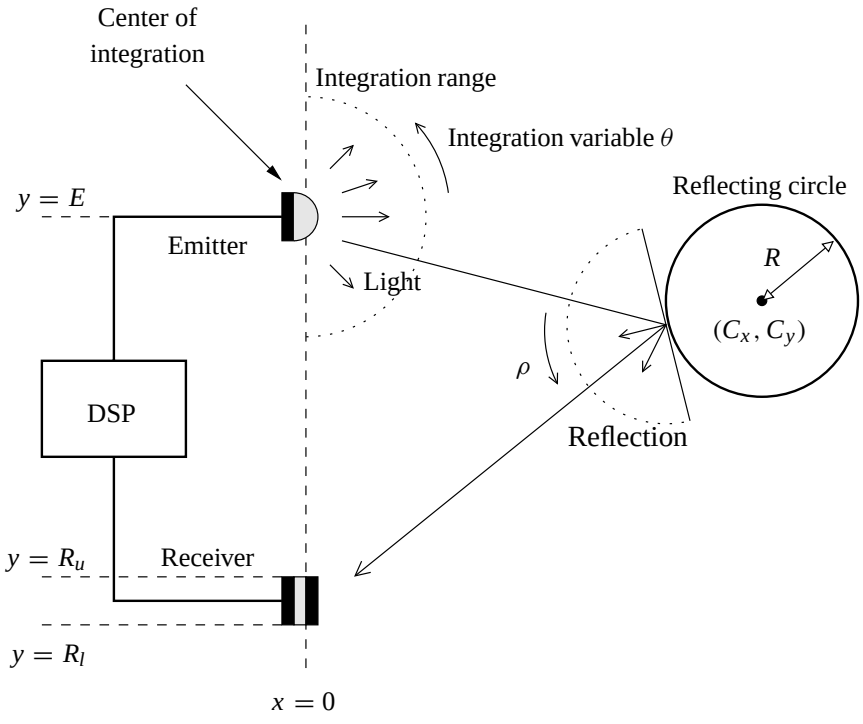


Figure 8.4: The basic components in the model.

The radius vector added to the center vector yields the point

$$\big(C_x - R\beta(1+\beta^2)^{-1/2},\ C_y + R(1+\beta^2)^{-1/2}\big).$$

The second slope α_2 is then given as

$$\alpha_2 = \frac{C_y + R(1+\beta^2)^{-1/2} - E}{C_x - R\beta(1+\beta^2)^{-1/2}} = \frac{(C_y - E)\sqrt{1+\beta^2} + R}{C_x\sqrt{1+\beta^2} - R\beta}.$$

The integration interval is $[\arctan\alpha_1;\ \arctan\alpha_2]$.

8.2.2 Reflection onto the Receiver

Since the integration variable is the angle (and not the slope) an equidistantly discretization of the integral interval ensures that the amount of emitted light (the directional characteristic of the emitter disregarded) is the same for any discrete angle in the integration interval. For each discrete angle the emitted light covers a small interval on the circle. This interval will be considered straight, since this allows it to be described by the surface

model presented in Section 8.1.3. The distribution of the light reflected by this small area is then determined by $m(\rho)$ and the angle of incidence φ. For any given angle $\theta \in [\theta_1;\ \theta_2]$ the light arrives at the circle in a point P determined as the intersection of the circle and the line through $(0, E)$ with slope $\tan\theta$. Knowing this point the angle of incidence is easily found.

Given the distribution of the reflected light the amount of light reflected onto the receiver is found by considering the small area as a point, since this gives the small area the same properties as an emitter, i.e. the total amount of light 'emitted' within an angle interval, as covered by the receiver (e.g. $[\rho_1; \rho_2]$ in Fig. 8.5) is found by integration. Since the reflecting point P is known, it is easy to determine this interval. The entire setup is depicted in Fig. 8.5.

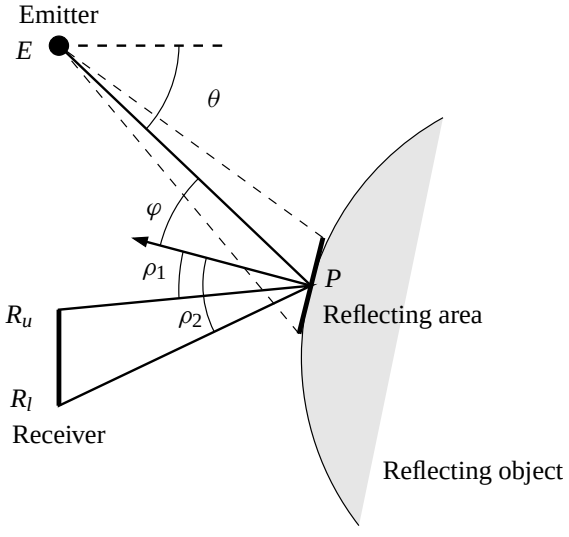


Figure 8.5: The light emitted at (the discrete) angle θ falls on a small area, which reflects the light onto the receiver. For a given θ the quantities φ, ρ_1, ρ_2, and P are determined by geometrical calculations.

First the point P is found. The points of intersection of the line with slope $\alpha = \tan\theta$ and the circle

$$(0, E) + t(1, \alpha), \qquad (x - C_x)^2 + (y - C_y)^2 = R^2,$$

is determined by those t which solves

$$(1 + \alpha^2)t^2 + 2(\alpha(E - C_y) - C_x)t + (C_y - E)^2 + {C_x}^2 = R^2.$$

The smallest root

$$t = \frac{\alpha(C_y - E) + C_x - \sqrt{2C_x(C_y - E)\alpha + (R^2 - C_x^2)\alpha^2 - (C_y - E)^2 + R^2}}{1 + \alpha^2}$$

inserted into the line yields P. The explicit expression is omitted, and the coordinates to the point is denoted (P_x, P_y).

The angle of incidence φ is found as the angle between the 'slope vector' $[1, \alpha]^\top$ and radius vector $[C_x - P_x; C_y - P_y]^\top$. This would usually be

$$\varphi = \arccos \frac{\begin{bmatrix} 1 \\ \alpha \end{bmatrix}^\top \begin{bmatrix} C_x - P_x \\ C_y - P_y \end{bmatrix}}{R\sqrt{1 + \alpha^2}},$$

but this formula will always give a positive number. In order to give φ the right sign (which dependents on whether the incoming light is 'above' or 'below' the normal to the circle) the following is used instead.

$$\begin{aligned} \varphi &= \arccos \frac{\begin{bmatrix} -\alpha \\ 1 \end{bmatrix}^\top \begin{bmatrix} C_x - P_x \\ C_y - P_y \end{bmatrix}}{R\sqrt{1 + \alpha^2}} - \frac{\pi}{2} \\ &= \arccos \frac{C_y - P_y + (P_x - C_x)\alpha}{R\sqrt{1 + \alpha^2}} - \frac{\pi}{2}. \end{aligned} \tag{8.3}$$

The two angles ρ_1 and ρ_2 are found in a similar fashion. The formula for ρ_1 is

$$\begin{aligned} \rho_1 &= \arccos \frac{\begin{bmatrix} P_y - R_u \\ -P_x \end{bmatrix}^\top \begin{bmatrix} P_x - C_x \\ P_y - C_y \end{bmatrix}}{R\sqrt{{P_x}^2 + (R_u - P_y)^2}} - \frac{\pi}{2} \\ &= \arccos \frac{(P_y - R_u)(P_x - C_x) - P_x(P_y - C_y)}{R\sqrt{{P_x}^2 + (R_u - P_y)^2}}, \end{aligned}$$

while the formula for ρ_2 is identical except that R_u is substituted for R_l.

There is, however, another way to calculate φ which comes in handy when interpreting the integral equation. From Fig. 8.6 it follows that

$$\frac{r}{\sin(\pi - \varphi)} = \frac{R}{\sin(\theta - \theta_c)} \quad \Leftrightarrow \quad \varphi = \pi - \arcsin \frac{r \sin(\theta - \theta_c)}{R}.$$

and when rewriting to obtain the correct sign

$$\varphi = \arcsin \frac{r \sin(\theta - \theta_c)}{R}. \tag{8.4}$$

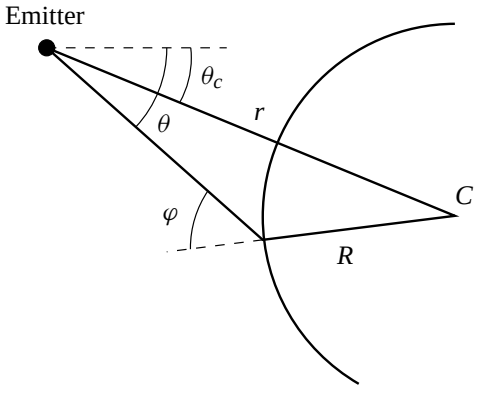


Figure 8.6: An alternative way to find φ.

Since this equation is in polar coordinates, like the integral equation in the next section, it is the one to use when interpreting the integral equation. Notice, however, that the measurements presented in Section 8.3.1 are in Cartesian coordinates. It is therefore necessary when implementing the integral equation to either changes the coordinates in (8.4) using

$$r = \sqrt{{C_x}^2 + (C_y - E)^2}, \quad \theta_c = \arctan \frac{C_y - E}{C_x}$$

or to use (8.3) instead. Although this form includes the point P, which therefore has to be calculated, it does not increase the number of calculations since this point is needed anyway (to find ρ_1 and ρ_2).

8.2.3 Formulating the Model as an Integral Equation

Having modeled the reflection from the surface with (8.1), the angle of incidence with (8.4), and by letting I_e be the directional characteristic of the emitter, it is now possible to form the equation which models the entire setup.

The entire amount of light reflected by the circle onto the receiver is the sum of the amounts reflected by the small areas which each corresponds to a value of the discrete angle θ. For infinitely small θ this sum becomes an integral.

$$I(\theta_c, r) = \int_{\theta_1}^{\theta_2} I_e(\theta) \int_{\rho_1}^{\rho_2} m\Big(\rho - \arcsin \frac{r \sin(\theta_c - \theta)}{R}\Big) d\rho d\theta, \tag{8.5}$$

where m is any sufficiently nice function (determined by the surface properties of the circle), θ_c and r is the center (polar) coordinate of the circle, and

$$\theta_1 = \arctan \frac{EC_x - C_yC_x + R\sqrt{{C_x}^2 - R^2 + (C_y - E)^2}}{R^2 - {C_x}^2},$$

$$\theta_2 = \arctan \frac{(C_y - E)\sqrt{1+\beta^2} + R}{C_x\sqrt{1+\beta^2} - R\beta},$$
$$\beta = \frac{R_l C_x - C_y C_x - R\sqrt{{C_x}^2 - R^2 + (C_y - R_l)^2}}{R^2 - {C_x}^2},$$
$$\rho_1 = \arccos \frac{(P_y - R_u)(P_x - C_x) - P_x(P_y - C_y)}{R\sqrt{{P_x}^2 + (R_u - P_y)^2}},$$
$$\rho_2 = \arccos \frac{(P_y - R_l)(P_x - C_x) - P_x(P_y - C_y)}{R\sqrt{{P_x}^2 + (R_l - P_y)^2}}.$$

The point P, that is (P_x, P_y), is found by inserting

$$t = \frac{\alpha(C_y - E) + C_x - \sqrt{2C_x(C_y - E)\alpha + (R^2 - C_x^2)\alpha^2 - (C_y - E)^2 + R^2}}{1+\alpha^2}$$

into the line $(0, E) + t(1, \alpha)$, there $\alpha = \tan\theta$, θ being the integration variable for the outermost integral. Moreover

$$C_x = r\cos(\theta_c), \quad C_y = r\sin\theta_c.$$

The quantities R, E, R_l, and R_u are all constants. The visualization of the function $I(\theta_c, r)$ for a particular circle produces a three dimensional map, which will be denoted the *intensity map* for that circle.

8.2.4 Examples of Modeling

To demonstrate what type of 'output' the model (8.5) produces a number of examples are now given. The model has several adjustable parameters, and it would be rather extensive to explore all possible combination. The figures 8.7, 8.8, and 8.9 show six different choices of parameters. The position of emitter and receiver are fixed, however. The most remarkable observations is that for a flat emitter directional characteristic the position of the circle causing the highest intensity is very close to the receiver and quite far from the emitter. The cosine emitter characteristic used in the lowermost plot in Fig. 8.7 resembles the characteristic of the LED used for the measured data. The exponential emitter characteristic function used in the lowermost graph in Fig. 8.8 and both graphs in Fig. 8.9 have actually been found in one emitter.

To improve the speed of calculations (about 100 times) the innermost integral has been pre-calculated using

$$f(x) = \begin{cases} 0 & \text{for } x \in]-\infty; -\frac{\pi}{2}[, \\ \int_{-\frac{\pi}{2}}^{x} m(\rho)d\rho & \text{for } x \in [-\frac{\pi}{2}; \frac{\pi}{2}], \\ f(\frac{\pi}{2}) & \text{for } x \in]\frac{\pi}{2}; \infty[, \end{cases} \tag{8.6}$$

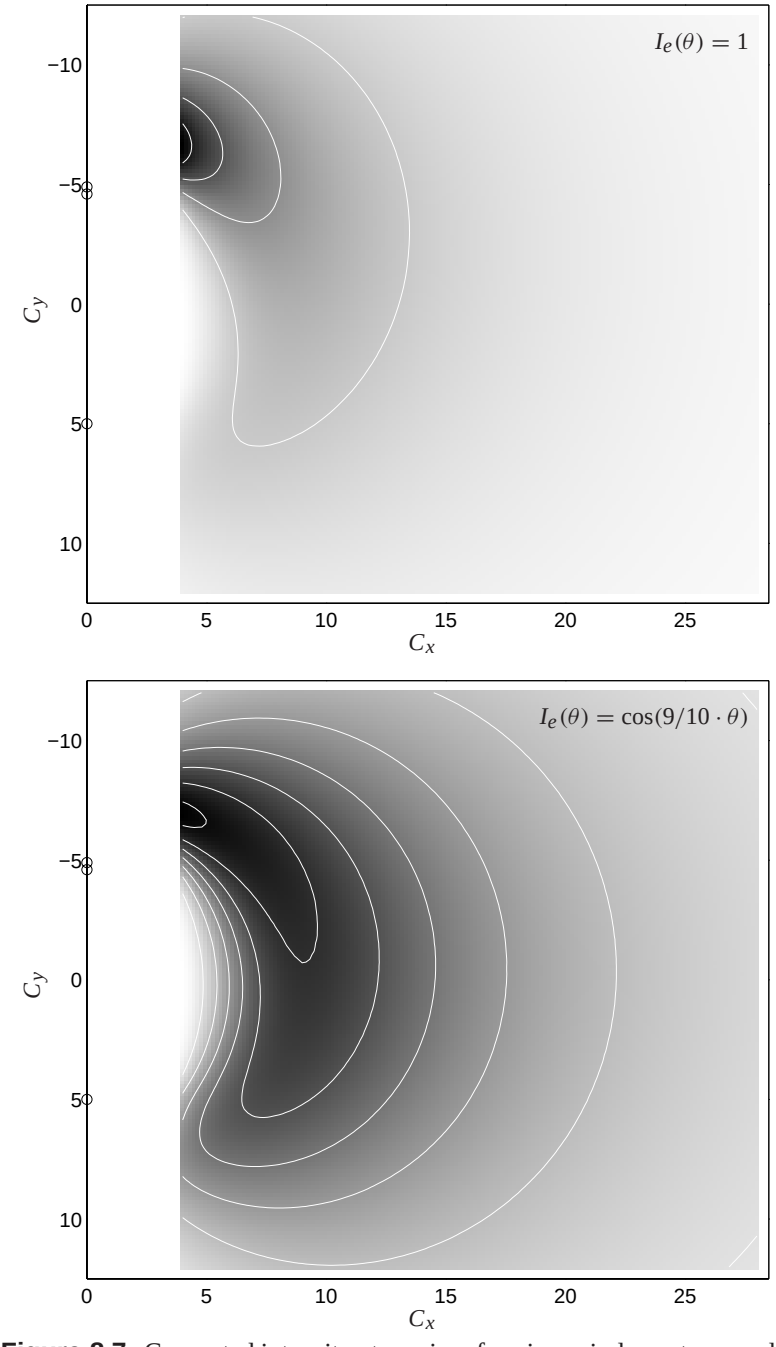


Figure 8.7: Computed intensity at receiver for given circle center coordinate. In both cases $R = 3$, $E = 5$, $R_u = -4.9$, and $R_l = -5.1$.

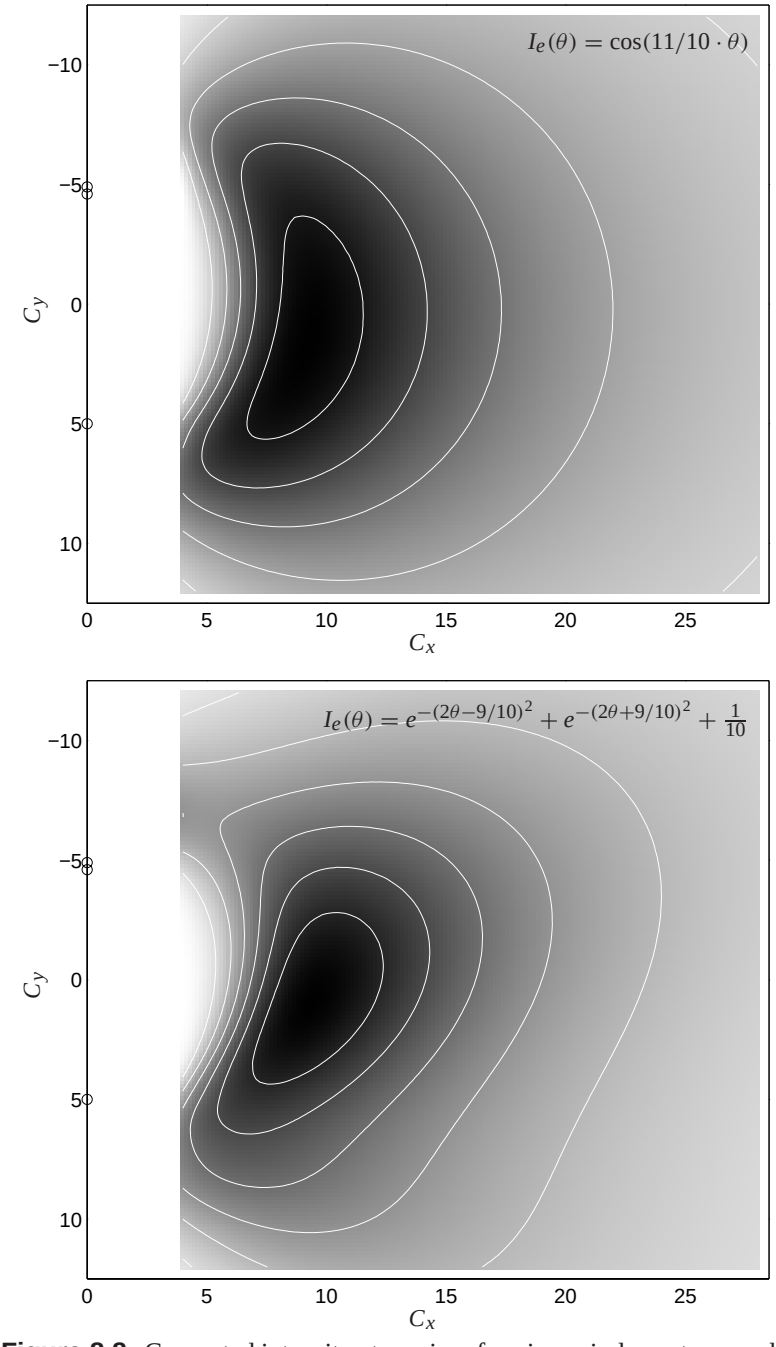


Figure 8.8: Computed intensity at receiver for given circle center coordinate. In both cases $R = 3$, $E = 5$, $R_u = -4.9$, and $R_l = -5.1$.

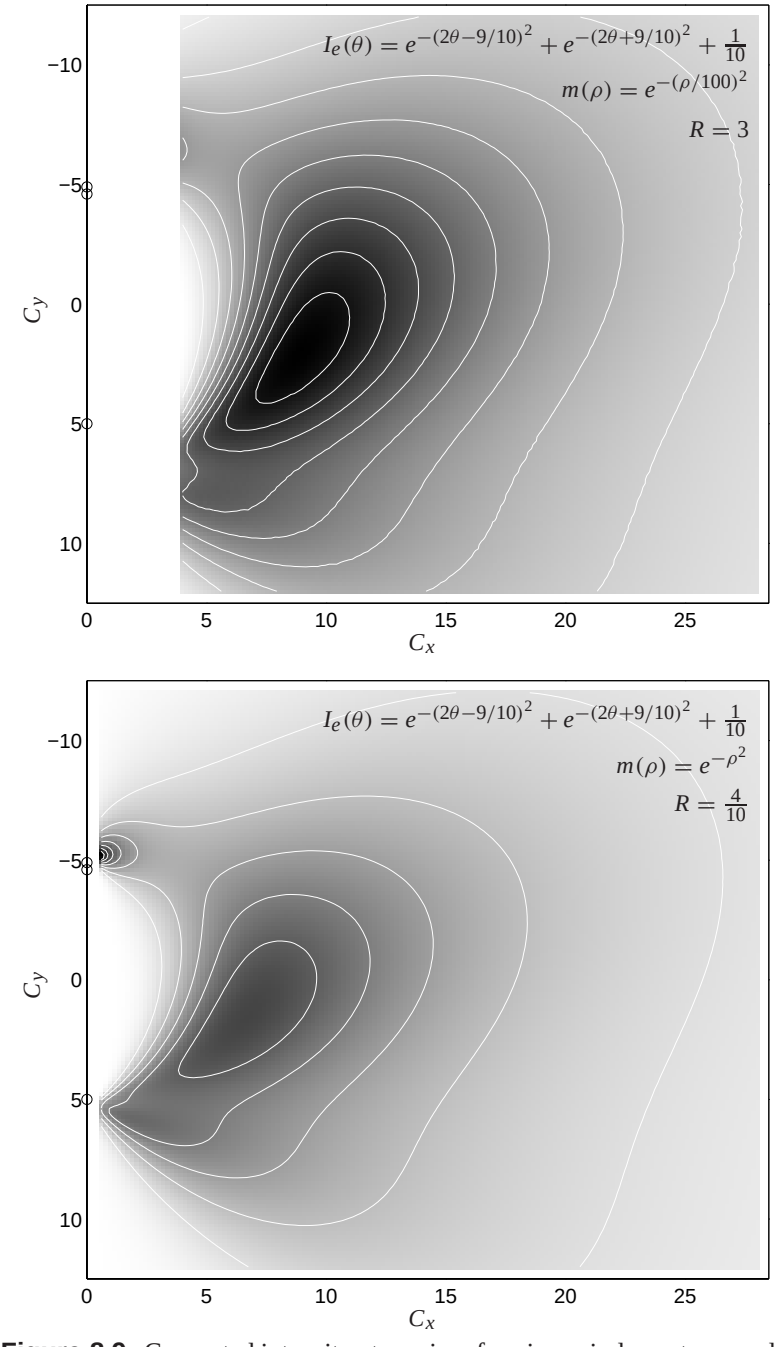


Figure 8.9: Computed intensity at receiver for given circle center coordinate. In both cases $E = 5$, $R_u = -4.9$, and $R_l = -5.1$.

which is sampled in 100.000 points in the interval $[-\pi;\pi]$. This is possible because the integrand in (8.1) is merely a translation ϕ of m and this corresponds to the same translation of f. Note that the integral of the functions used (Gaussian functions) is very small in the intervals $]-\infty;-\frac{\pi}{2}[$ and $]\frac{\pi}{2};\infty[$ making the approximation fairly accurate.

8.3 Evaluating the Model

The model which has been developed in the previous sections describes a two dimensional setup with an emitter and a receiver and a circular, diffuse-reflecting object. The output of the model is a reflection map that shows the reflected intensity of any given position of the reflecting object. Some assumptions were made to reduce the complexity of the model, and the next step after constructing the model is therefore to evaluate it, i.e. compare it to a real reflection map.

For this purpose an experimental setup has been made to provide the necessary data. This is described in Section 8.3.1. There are many possible means for comparing the measured and the modeled reflection maps. One such method, based on gradients, is presented in Section 8.3.2.

A more subtle approach to evaluating the model is using it for estimating the directional characteristics of the emitter by solving an inverse problem. This method is presented in Section 8.4 and 8.5.

8.3.1 Measuring a Real Reflection Map

The setup used for acquiring reflection map data is quite similar to the setups described in Chapter 5. The emitter is the same as used in the third setup, see Section 5.4.6. The receiver circuit is not described in Section 5, but it is quite similar to the one used in the second setup, in particular the photo diode is the same type. The WPT channel gain methods, as presented in Section 4.1.2 starting on page 32, is used to make the measurements. The signal length is 512 samples, sampled at 5 kHz. The reflection map data is an average over approximately one second (10 signals). A ball of light wood with diameter 60 mm has been used as reflecting object. It is mounted on a 300 mm long stick which is fixed on the head of an A3-size XY table. This enables the computer performing the data acquisition to control the XY position of the reflecting object. The accuracy of the table is < 0.01 mm. The setup is shown in Fig. 8.10.

Reflections has been measured in a grid with 31×70 points, which is equivalent to the physical rectangle $[50; 122] \times [-130; 70]$. The emitter is located at $(5, -45)$ and the receiver at $(5, 50)$. All units are mm. The measured reflections are shown in Fig. 8.11 with contour lines and a circle showing the size of the wooden ball. The same data is shown in three dimensions in Fig. 8.12. The height/coloring of the data is according to measured intensity which in this case is simply the inner product without any modifications. As argued in Section 4.6.1, page 55, this is a relative measure with any a priori physical

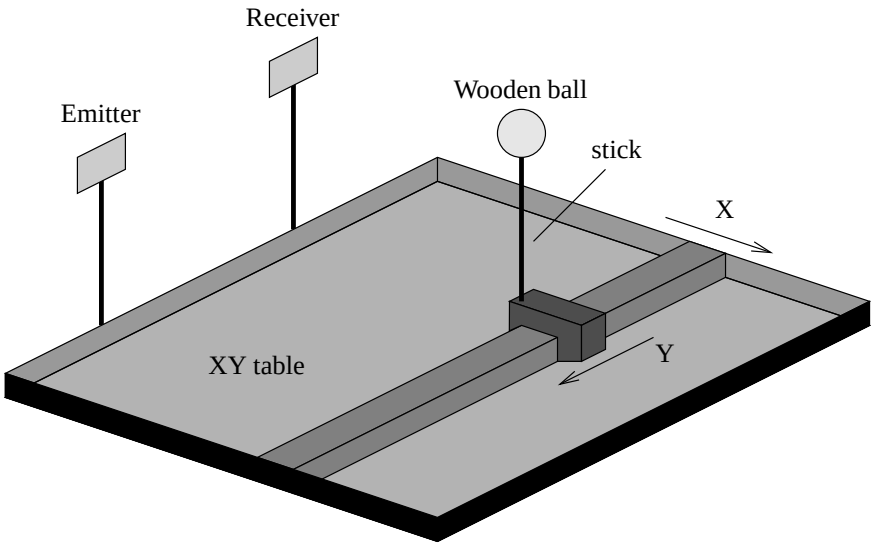


Figure 8.10: The setup with an XY table for measuring reflection map data. The X and Y motion corresponds to the axes on all the plots of the measured data.

interpretation, and no attempt has been made to related the amplitude of the data to the physical conditions of the setup.

The reflection map has an interesting structure which is highly asymmetric vertically and non-monotone horizontally. Both phenomena match poorly with the intuition; that the emitter and receiver are interchangeable and that the reflected intensity decreases when the reflecting object is moved further away from the emitter and receiver. The existence of these phenomena does not have any direct impact at this point. For now the main interest is determining the accuracy of the model. However, the three dimensional modeling of a multiple emitter/receiver setup presented in Chapter 7 relies heavily on the reflection map, in particular the geometrical structure of the isocandela curves.

8.3.2 Quality of Model Measured by Gradients

At a first glance the measured reflection map seems to exhibit the same structure as the model when emitter and receiver is assumed to have cosine-like directional characteristics, see the lowermost reflection map in Fig. 8.7. However, a closer examination reveals that there is a mismatch in terms of variation in intensities as well as the directions of the isocandela curves.

There are several adjustable parameters in the model and it is possible to find the best match simply by varying all the parameters in an exhaustive search for the optimum. An important question in the respect is what should be the measure for the goodness of a

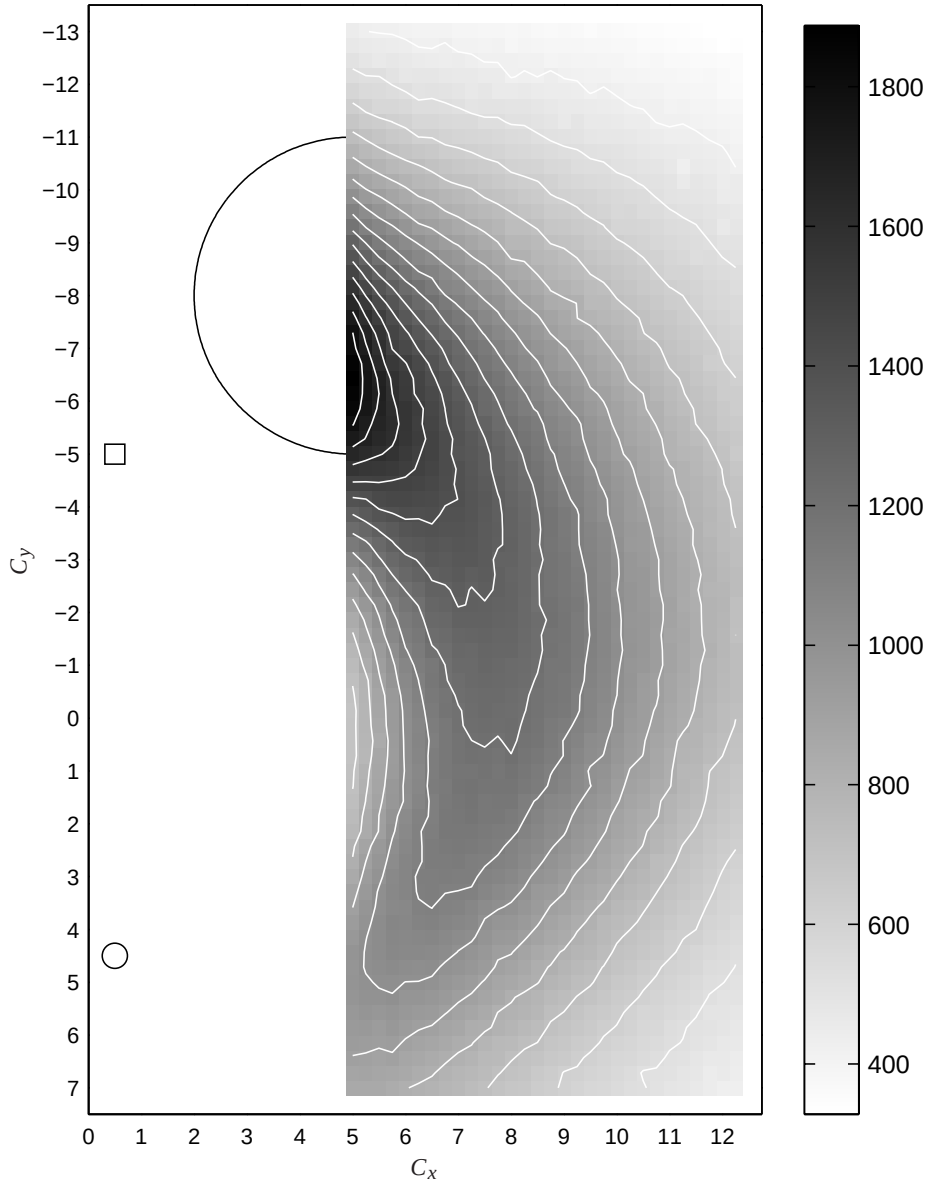


Figure 8.11: The measured reflection map. The small circle shows the position of emitter and the small square shows the position receiver. The big circle shows the size of the reflecting wooden ball. The white lines shows 15 evenly distributed contour lines (isocandela curves). The axes unit is cm.

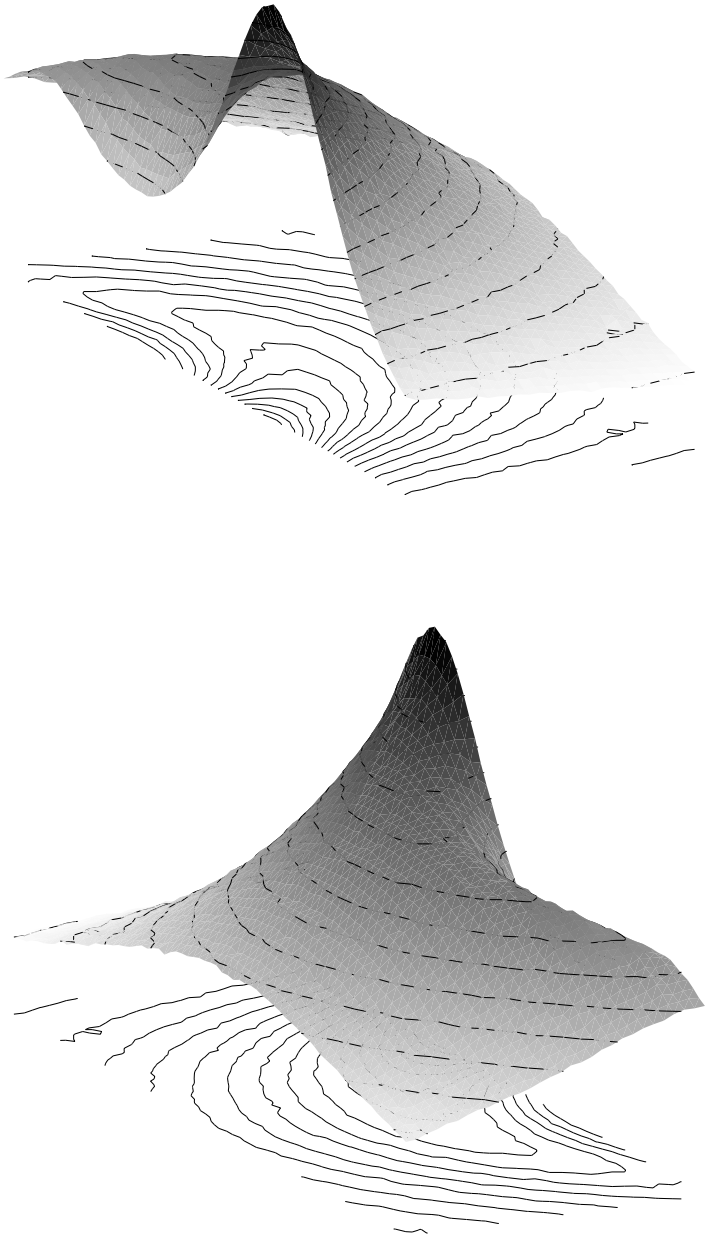


Figure 8.12: A three dimensional view from two different angles of the measured reflection map with isocandela curves superimposed.

match. The perhaps most apparent measure is the ℓ^2 norm of the difference between the two reflection maps, but experiments will show that the ℓ^2 norm optimum is a model with isocandela curves quite different from the ones of the measured data and with parameters that in some cases do not come close to the real value of the parameter.

An alternative measure is therefore the difference between isocandela curves. This measure can be defined as the total sum of deviations between the projections of the gradients onto the xy plane in the two maps in each point, i.e. the angular deviation in the xy coordinates between the normal vectors to the tangent planes in each point. The projection onto the xy plane is not mandatory, but it makes the measure insensitive to difference in amplitudes/scaling of the two maps. In the discrete case the tangent plane must be based on the number of neighboring points, and in the present case the four adjacent points are used. An exhaustive search with varying parameters yields a result (not shown here) which is much closer to the measured data than the particular model presented in Fig. 8.7. However, there is still a significant difference which seems to originate in the structure of the model rather than the parameters. Recalling that the model is two dimensional and the measurements are acquired in three dimensions, it seems worth to attempt to adjust the model to include the third dimension. Instead of redoing all the geometrical observations and derivations in three dimensions a simple 'compensation' can be applied. The extra dimension can crudely be added by multiplying the reflection map in each point by a factor that is inversely proportional to the distance from emitter to the point and to the receiver raised to the power n, where $n = 1$ is the natural choice. The result of this action is shown in Fig. 8.13. The left plot shows the isocandela curves for the measured and the modeled reflection maps, while the right plots shows the angular difference measure. Parallel isocandela curves means zero difference (white) and orthogonal isocandela curves means maximum difference (black). Notice how the difference is largest where the curves has the highest curvature. This is because even a small misalignment of the curves in this particular case produces large angular differences. There also seems to be a measurement error in the top right corner of the data set. The parameters corresponding to the best match is given in Table 8.1.

Table 8.1: Parameters for real data and best model.

	Reality	Model
Y location of emitter	-45 mm	-43.3 mm
Y location of receiver	50 mm	-47.5 mm
Extension of receiver	2.2 mm	1.67 mm
Radius of object	30 mm	31.3 mm
Reflection model of object	N/A	$e^{-\lvert\theta\rvert^3}$
Power on 3rd dim. compensation	N/A	1.20

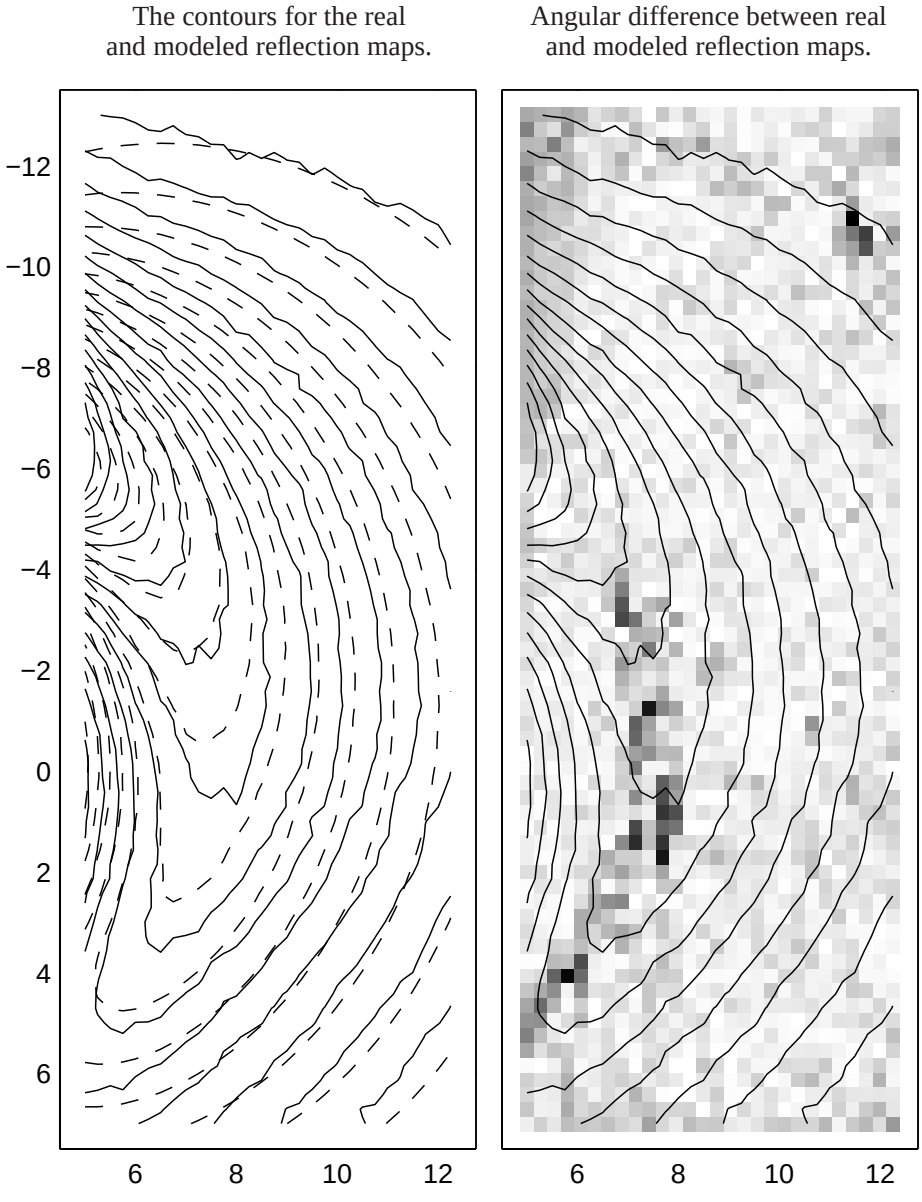


Figure 8.13: A visual estimation of the accuracy of the reflection map model. The left plot shows 15 evenly distributed contour lines, i.e. isocandela curves, for the measured (solid) and modeled (dashed) reflection maps. The right plot shows the same contour lines for the measured reflection map and the gray-shaded squares show the angular difference for isocandela curves in each point for the measured and modeled reflection map. White is zero difference, and black is for orthogonal curves.

8.4 Solving the Integral Equation

The evaluation of the model in the previous section focused on the immediate relation between model and reality. Although this gives a qualified hint as to whether the model is good or completely off track it does not reveal the whole truth about the model. In particular the model might be more accurate at some xy areas than others, and the slightly 'wrong' parameters in Table 8.1 might have some so far undiscovered effect on the model. Obviously, it would be nice to have another estimate of the model accuracy, one which is not based on parameter optimization. The that end it might be beneficial to regard the model as an inverse problem $\mathbf{Ax} = \mathbf{b}$ where the mapping $\mathbf{A} \sim I(\theta_c, r)$ is the model, the output $\mathbf{b}$ is the measured data, and the input $\mathbf{x} \sim I_e(\theta)$ is an unknown emitter characteristic. This means going from the measurements via the model back to the emitter characteristic. In reality the emitter characteristic is known since it has been measured, and comparing the the true characteristic (also called the true solution) to the solution of the inverse problem may indicate how accurate the model is.

The measured data does not come in a infinitely fine resolution, so the integral equation must be discretized and the problem then converts to a matrix inversion problem with an ill-conditioned matrix. Solving the matrix equation by brute force is therefore prone to produce a numerically unstable solution, and regularization methods are required to obtain a reasonable solution. The discretization is described in the next section, while the regularization is described in Section 8.5.

All what remains of this chapter is based on theory of numerical deconvolution. In particular, the preprint [36] and a regularization toolbox [37], both by Hansen, has been very useful. Thanks are also due to Hansen himself for reading this part of the chapter and providing useful suggestions.

8.4.1 Discretization of the Integral Equation

To identify the model (8.5) as an inverse problem it is helpful to rewrite it into

$$I(\theta_c, r) = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} K(\theta_c, r, \theta) I_e(\theta) d\theta, \tag{8.7}$$

where the kernel K is

$$K(\theta_c, r, \theta) = \int_{\rho_1(\theta_c, r)}^{\rho_2(\theta_c, r)} m\Big(\rho - \arcsin \frac{r \sin(\theta_c - \theta)}{R}\Big) 1_{[\theta_1(\theta_c, r); \theta_2(\theta_c, r)]} d\rho.$$

The form (8.7) reveals the model to be a Fredholm integral equation of the first kind. As just described only discrete solutions are interesting, and a Fredholm integral equation can be discretized into a set of linear equations on the form

$$\sum_{n=1}^{N} w_n K(\theta_{c,m}, \theta_n) I_e(\theta_n) = I(\theta_{c,m}, r), \quad m = 1, \ldots, M,$$

where $\theta_{c,m},\ \theta_n \in]-\frac{\pi}{2};\frac{\pi}{2}[$. The quantity w_n is a weight parameter determined by the method of discretization (trapeze, Simpson etc.). In terms of matrices the model is written $\mathbf{Ax} = \mathbf{b}$ with

$$\left.\begin{array}{c} a_{mn} = w_n K(\theta_{c,m}, \theta_n, r) \\ x_n = I_e(\theta_n) \\ b_m = I(\theta_m) \end{array}\right\} \qquad \begin{array}{l} n = 1, \ldots, N, \\ m = 1, \ldots, M. \end{array}$$

The **A** matrix is always a band matrix in the sense that the left uppermost and right lowermost parts are zeros. The band itself can be curved (if for instance the circle is moved along a line in Cartesian coordinates) or straight if the distance r to the circle is fixed. The former type is useful when the intensity map is sampled in a Cartesian grid. The

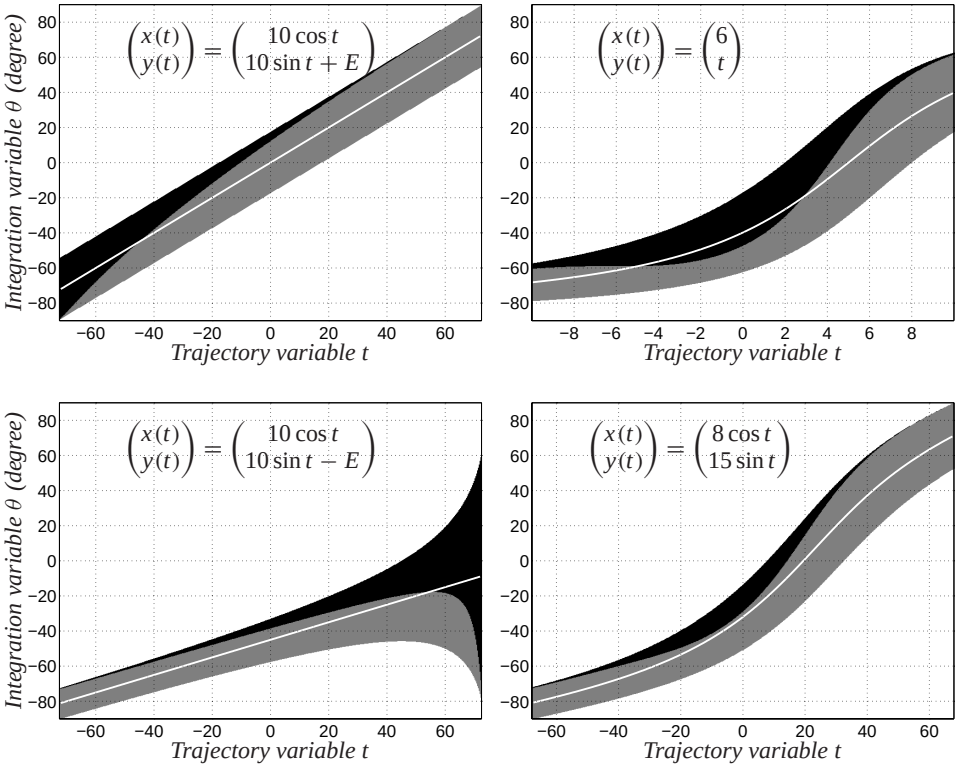


Figure 8.14: The gray area shows at what angle interval there is potential reflection from the circle (visible from receiver), while the black area shows the angle interval at which the circle is illuminated but not visible from the receiver. The parameter functions describe the center coordinate of the circle. The white line shows the corresponding θ_c which is always in the middle of the bands.

first form used in Section 8.5.2 which describes the results of regularization. The generic kernel matrices **A** for four different forms are shown in Fig. 8.14. The horizontal width of the band corresponds to the size of the circle as seen from the emitter.

Since the equations presented in the previous sections does not support a position of the circle where it intersects $x = 0$ the variable θ_c must be in the interval $[-\frac{\pi}{2} + v; \frac{\pi}{2} - v]$, where $v = \arcsin(R/r)$ is half the size of the circle as seen from the emitter. The end points of this interval are reached when the circle 'touches' $x = 0$. The integration variable θ will always cover a half circle, though.

8.4.2 Need for Regularization

The first solution method applied to the matrix problem $\mathbf{Ax} = \mathbf{b}$ is the straight forward $\mathbf{x} = \mathbf{A}^{\dagger}\mathbf{b}$ († meaning pseudo inverse). This method will do in some cases, namely when the matrix is not ill-conditioned and the measurements **b** are noise-free. Neither of these conditions are fulfilled for the inverse problem at hand, and the direct solution clearly shows that a more subtle approach should be used. This approach is shown in Fig. 8.15. The measurements **b** and the model generated equivalent **Ax** are shown in the first plot. They seem to correspond fairly well. The true solution **x** and the band matrix is also shown separately. The matrix is 100×100. The non-vanishing parts corresponds to the gray parts of the band matrices shown in Fig. 8.14. When **A** is (pseudo) inverted and multiplied with **b** the result is highly erratic. The problem is the high condition number for the matrix combined with a noisy **b**. If the condition numbers is determined by the largest singular value divided by the smallest, the **A** matrix in this case, the one shown in Fig. 8.15, has a condition number in the order of 10^{21}! This follows from the plot of the singular values in the same figure.

8.5 Singular Value Decomposition Solution Approach

Their exist a number of different approaches for solving ill-conditioned inverse problems. One of these is the SVD approach which utilizes the decomposition of the matrix into orthogonal matrices and singular values. To see how this can improve the regularity of a solution some linear algebra is needed.

8.5.1 Basic SVD Theory

The SVD is defined for any $m \times n$ matrix **A** and takes the form

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top} = \sum_{k=1}^{\min(m,n)} \mathbf{u}_k \sigma_k \mathbf{v}_k^{\top}, \quad \mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_m], \quad \mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n].$$

Both decomposition matrices **U** and **V** are orthogonal. This implies that the singular vectors $\mathbf{u}_k$ form an orthonormal set, and likewise for $\mathbf{v}_k$. The middle matrix $\mathbf{\Sigma}$ is a diagonal

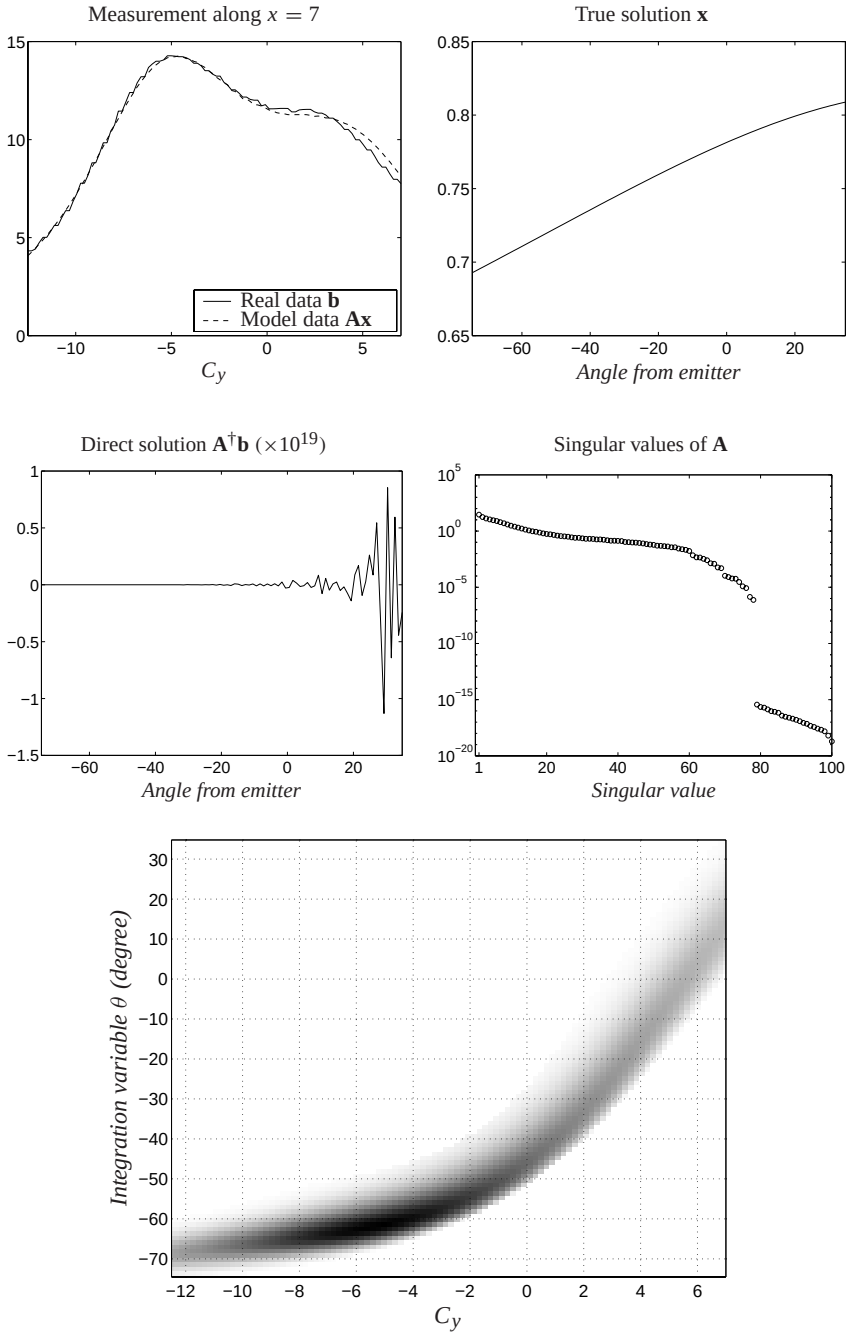


Figure 8.15: The components in the inverse problem. The individual plots are presented in the text. Note that the direct solution is actually obtained by Gaussian elimination and not via the pseudo inverse, and note that the true solution is the measured directional characteristic of the emitter.

matrix whose diagonal elements σ_k are the singular values in non-increasing order. The condition number of $\mathbf{A}$ has a simple expression in terms of the SVD if the 2-norm is used, because

$$\text{cond}(\mathbf{A}) = \|\mathbf{A}\|_2\|\mathbf{A}^\dagger\|_2 = \frac{\sigma_1}{\sigma_{\min(m,n)}}, \quad \|\mathbf{A}\|_2 = \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2},$$

i.e. $\|\mathbf{A}\|_2$ is the operator norm of $\mathbf{A}$. Since the column space of $\mathbf{u}_k$ is $\mathbb{R}^m$ and the column space of $\mathbf{v}_k$ is $\mathbb{R}^n$, the vectors $\mathbf{b}$ and $\mathbf{x}$ can be decomposed in these bases

$$\mathbf{b} = \sum_{k=1}^{m} \langle \mathbf{u}_k, \mathbf{b}\rangle \, \mathbf{u}_k = \sum_{k=1}^{m} (\mathbf{u}_k^\top \mathbf{b})\mathbf{u}_k \qquad \text{and} \qquad \mathbf{x} = \sum_{k=1}^{n} (\mathbf{v}_k^\top \mathbf{x})\mathbf{v}_k.$$

Then

$$\mathbf{A}\mathbf{x} = \sum_{k=1}^{n} (\mathbf{v}_k^\top \mathbf{x})\mathbf{A}\mathbf{v}_k = \sum_{k=1}^{\min(m,n)} (\mathbf{v}_k^\top \mathbf{x})\sigma_k \mathbf{u}_k.$$

The last equation is derived from the property $\mathbf{A}\mathbf{v}_k = \sigma_k \mathbf{u}_k$ of the SVD. Noting that

$$(\mathbf{u}_k^\top \mathbf{b}) = (\mathbf{v}_k^\top \mathbf{x})\sigma_k \quad \Leftrightarrow \quad \frac{(\mathbf{u}_k^\top \mathbf{b})}{\sigma_k}\mathbf{v}_k = (\mathbf{v}_k^\top \mathbf{x})\mathbf{v}_k, \quad k = 1, \ldots, n,$$

the equation $\mathbf{A}\mathbf{x} = \mathbf{b}$ can be rewritten to (now for simplicity assuming $m \geq n$)

$$\sum_{k=1}^{n} \frac{(\mathbf{u}_k^\top \mathbf{b})}{\sigma_k}\mathbf{v}_k = \sum_{k=1}^{n} (\mathbf{v}_k^\top \mathbf{x})\mathbf{v}_k = \mathbf{x}. \tag{8.8}$$

This equation provides the brute force solution to the original problem $\mathbf{A}\mathbf{x} = \mathbf{b}$. The major difference compared to the solution $\mathbf{A}^\dagger\mathbf{b}$ is that (8.8) clearly demonstrates a potential numerical instability in containing a fraction of which the denominator is a non-increasing sequence. In practice the instability is unavoidable; the following three properties for a matrix $\mathbf{A}$ that arises from the discretization of first-kind Fredholm integral equation and the concluding arguments demonstrates this.

1. The singular values of $\mathbf{A}$ decay gradually. They level off, however, if machine precision is reached.
2. The singular vectors $\mathbf{u}_k$ and $\mathbf{v}_k$ have an increasing number of sign changes in their elements as k increases. Often, the number of sign changes is precisely $k - 1$.
3. Whenever there exist a solution $f \in L^2([-\frac{\pi}{2}; \frac{\pi}{2}])$ to the Fredholm integral equation the quantities $|\mathbf{u}_k^\top \mathbf{b}|$ will decay faster than the singular values, until they reach a plateau approximately equal to the noise level in $\mathbf{b}$ at which they level off.

An immediate consequences is that in the region in which $|\mathbf{u}_k^\top \mathbf{b}|$ is decreasing the coefficients $|\mathbf{u}_k^\top \mathbf{b}|/\sigma_k$ are also decreasing. The sum of $(\mathbf{u}_k^\top \mathbf{b}/\sigma_k)\mathbf{v}_k$ in this region is dominated by slow oscillations. But whenever $|\mathbf{u}_k^\top \mathbf{b}|$ levels off the singular values keep decreasing,

and in this region the coefficients $|\mathbf{u}_k^\top \mathbf{b}|/\sigma_k$ are increasing and might easily reach a level above the starting level for the coefficients. Since in this region $\mathbf{v}_k$ has an increasing number of oscillations the sum of $(\mathbf{u}_k^\top \mathbf{b}/\sigma_k)\mathbf{v}_k$ in this region is dominated by fast oscillations, the amplitude of which is often high compared to the slowly oscillating part of $\mathbf{x}$.

The described properties and their effect are demonstrated in Fig. 8.15 and 8.16. In the latter the singular values are plotted along with the quantities $|\mathbf{u}_k^\top \mathbf{b}|$ and the coefficients $|\mathbf{u}_k^\top \mathbf{b}|/\sigma_k$. The singular values are gradually decaying all the way through, while $|\mathbf{u}_k^\top \mathbf{b}|$ is decaying until the 35th element after which they level off at about $5 \cdot 10^{-2} = 0.05$. The plot of $\mathbf{b}$ and $\mathbf{Ax}$ in Fig. 8.15 show an approximate noise level with the same order of magnitude, in agreement with the third property.

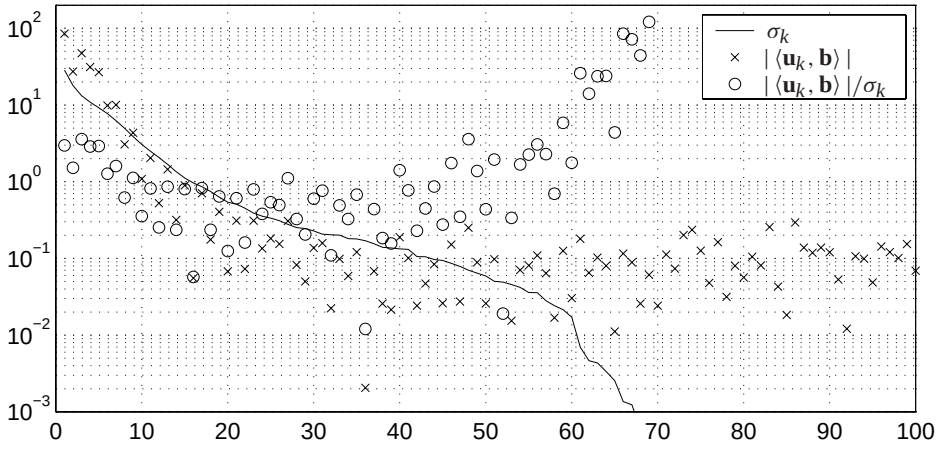


Figure 8.16: Picard plot.

If $K : \mathbb{R}^2 \mapsto \mathbb{R}$ is in $L^2([-\frac{\pi}{2}; \frac{\pi}{2}])$ and

$$b(s) = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} K(s,t)x(t)dt,$$

then there exist functions $u_k(s)$ and $v_k(t)$, $k \in \mathbb{N}$, such that

$$K(s,t) = \sum_{k \in \mathbb{N}} \mu_k u_k(s) v_k(t)$$

and both $\{u_k\}_{k\in\mathbb{N}}$ and $\{v_k\}_{k\in\mathbb{N}}$ are orthonormal basis for $L^2([-\frac{\pi}{2}; \frac{\pi}{2}])$. This decomposition is called singular value expansion (SVE). It is easy to derive (in a fashion similar to the one for SVD)

$$x(t) = \sum_{k \in \mathbb{N}} \frac{\langle u_k, b \rangle}{\mu_k} v_k(t). \tag{8.9}$$

The relation

$$\sum_{k\in\mathbb{N}} \left| \frac{\langle u_k, b\rangle}{\mu_k} \right| < \infty$$

is a necessary and sufficient condition for (8.9) to hold. In the context of inverse problems this inequality is also known as the Picard condition. This explains why the plot in figure 8.16 is called a Picard plot.

8.5.2 Truncated SVD and Tikhonov Regularization

Having identified the cause of the instability of the direct solution the next step is to suggest some method to reduce or maybe even eliminate the instability. Based on the observations made in the previous section it is natural to start by reducing the number of addends in the sum (8.8) by excluding a number of the terms with the highest indices. The choice of truncation parameter is quite easy: One of the conclusions in the previous section is that the SVD components $(\mathbf{u}_k^\top \mathbf{b}/\sigma_k)\mathbf{v}_k$ in (8.8) can be 'trusted' as long as $|\mathbf{u}_k^\top \mathbf{b}|$ is decreasing. By inspection of Fig. 8.16 it therefore seems that the first 34 to 40 terms would be appropriate. This rather brute method of regularization is denoted truncated SVD (TSVD).

The result of using this on the problem at hand is shown in Fig. 8.17. The slow oscillations are clearly dominating the solutions consisting of only a few terms, and as the number of terms increases so does the irregularity, especially in the left part of the solution curve. The result of the TSVD is not really satisfying as the oscillations dominating the solution for any number of sum terms.

The TSVD solution $\mathbf{x}_K$ for the truncation parameter K is equal to $\mathbf{A}_K^\dagger \mathbf{b}$ where

$$\mathbf{A}_K = \sum_{k=1}^{K} \frac{\mathbf{u}_k^\top \mathbf{b}}{\sigma_k} \mathbf{v}_k.$$

This solution also solves the minimization problem

$$\min \|\mathbf{x}\|_2 \quad \text{subject to} \quad \min \|\mathbf{A}_K \mathbf{x} - \mathbf{b}\|_2.$$

Note that the latter minimization problem has an infinity of solutions (since $\text{rank}(A_K) = K$), and the one with minimum 2-norm is singled out. Based on this observation a more subtle approach is now conceivable. By abandoning the desire to eliminate the residual norm $\|\mathbf{A}_K \mathbf{x} - \mathbf{b}\|$ a smaller solution norm is achievable. By combining the two minimization requirements into

$$\min\left\{ \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda^2 \|\mathbf{x}\|_2^2 \right\}$$

a trade off, controlled by the regularization parameter λ, between residual and solution norm is possible. This method is most commonly referred to as Tikhonov regularization. It can be shown that the problem always has a unique solution, which is denoted the

Tikhonov solution. Note that for $\lambda \to 0$ the Tikhonov solution tends to the brute force solution, while the solution is smoothed out as $\mathbf{x} \to \mathbf{0}$ when $\lambda \to \infty$.

It can be shown that the SVD form of the Tikhonov solution is

$$\mathbf{x}_\lambda = \sum_{k=1}^{n} f_k \frac{\mathbf{u}_k^\top \mathbf{b}}{\sigma_k} \mathbf{v}_k, \quad \text{where } f_k = \frac{\sigma_k^2}{\sigma_k^2 + \lambda^2}. \tag{8.10}$$

Note that the TSVD solution also is on this form with $f_k \in \{0, 1\}$. The quantities f_k are called filter factors. They all satisfy $0 \le f_k \le 1$, and they control the damping of the individual SVD components. Specifically, if λ is fixed somewhere between σ_1 and σ_n, then for $\sigma_k \gg \lambda$ it follows from (8.10) that

$$f_k = \frac{\sigma_k^2 + \lambda^2 - \lambda^2}{\sigma_k^2 + \lambda^2} = 1 - \frac{-\lambda^2}{\sigma_k^2 + \lambda^2} \approx 1,$$

and for $\sigma_k \ll \lambda$ it follows that

$$\begin{aligned} f_k = \frac{\lambda^2 \sigma_k^2}{\lambda^2 \sigma_k^2 + \lambda^4} &= \frac{\lambda^2 \sigma_k^2 + \sigma_k^4}{\lambda^2 \sigma_k^2 + \lambda^4} - \frac{\sigma_k^4}{\lambda^2 \sigma_k^2 + \lambda^4} \\ &= \frac{1 + \sigma_k^2 \lambda^{-2}}{1 + \lambda^2 \sigma_k^{-2}} + O\Big(\frac{\sigma_k^4}{\lambda^4}\Big) = \frac{\sigma_k^2}{\lambda^2} + O\Big(\frac{\sigma_k^4}{\lambda^4}\Big) \approx \frac{\sigma_k^2}{\lambda^2}. \end{aligned}$$

This implies that the first SVD components, corresponding to the singular values greater than λ, contributes with almost full strength to the Tikhonov solution. Similarly, the last SVD components corresponding to singular values smaller than λ are damped considerably and therefore contribute very little to the solution. Hence, it is expectable to see the Tikhonov solution resembling the TSVD solution when K and λ are chosen such that $\sigma_K \approx \lambda$.

The Tikhonov solution to the problem is shown in Fig. 8.18. For λ around 2.5 the solution is not too bad, at least in the left half of the interval. Overall the Tikhonov solutions are somewhat better than the TSVD solutions. An even smoother solution is obtainable with the Tikhonov regularization (as seen for larger values of λ), but as λ increases so does the deviation from the true solution.

Another thing to notice is that all the solutions fail to resemble the true solution to the far left and in the right half of the interval. This may partly be due to the matrix $\mathbf{A}$ which (see Fig. 8.15) has less significant data in those particular ranges.

8.6 Conclusion

A model has been proposed for a setup of an emitter and a receiver facing in the same direction, and with a circular object reflecting the light from the emitter onto the receiver. The model includes the directional characteristics of the emitter and the receiver, and a

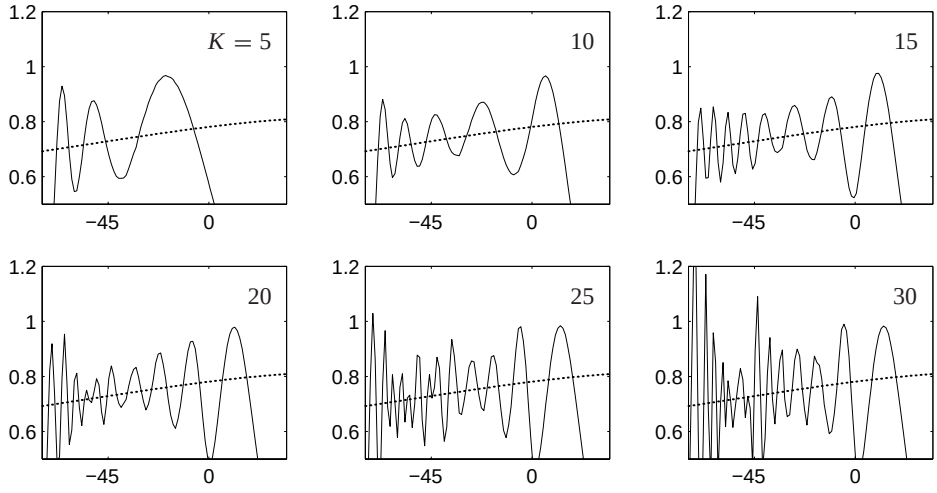


Figure 8.17: TSVD used on the problem in Fig. 8.15. The dotted curve is the true solution and the solid curve is the TSVD solution for the given number K of sum terms.

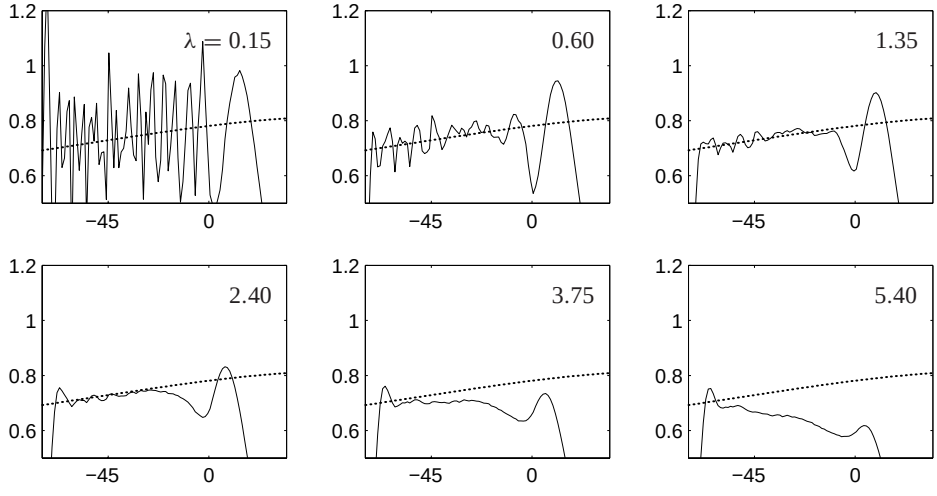


Figure 8.18: The Tikhonov solution for some values of λ. The dotted curve is the true solution.

description of the reflection property of the object. The model has been derived purely by geometrical observations and is formulated as a Fredholm integral equation of the first kind.

The output of the model is a reflection intensity map which shows the intensity of the reflected light for any (two dimensional) position of the reflecting object. A number of examples of such maps were given for a variety of parameter choices.

A physical setup was constructed in order to obtain a real reflection intensity map. Using ordinary emitter and receiver electronics, and an XY table for moving the reflecting object, a data set was recorded. An immediate comparison of the data set to the model revealed a structural difference which was crudely patch by introducing 'third dimension compensation'. This improved the model to an extent which called for a quantifiable comparison. Two methods were then applied to compare the modeled map to the measured map.

First the angular difference between gradients throughout the map was used a measure for evaluating the model. An exhaustive search in the parameter space yielded a rather good match, although the parameters did not match the real ones exactly. In this measure the model seems to be good, except in a small elongated region in the space between and in front of emitter and receiver. This measure essential compare isocandela curves, i.e. contour lines, of the maps.

An alternative method was applied to investigate the accuracy of the model. The model was considered an integral equation with unknown input, modeled kernel, and measured output. This yields an inverse problem since this approach is an attempt to go from the measured data via the model back to the directional characteristic of the emitter. Since this is known it is possible to quantify the model accuracy in an alternative fashion compared to the direct measure in the previously mentioned method. The modeled kernel is ill-conditioned, and thus the direct solution to the inverse problem is not useful. Applying Tikhonov regularization yields an estimated emitter characteristic which is fairly accurate in the region in front of the receiver. The other regions, i.e. in front of the emitter and on the other side of the receiver seems to be modeled less accurately.

The cause of this apparent inaccuracy has not been investigated thoroughly and is thus on the list of future work.

Part III

Wavelet and Rudin-Shapiro Transforms

9 The Problem of Finite Signals

One of the major issues in the field of applications wavelets is the handling of finite signals. The classical wavelet theory involving the multiresolution analysis (such as Daubechies [26]) is usually not concerned with this aspect. However, in most signal processing applications using wavelets this issue is of importance, since all real world signals are obviously finite. In some cases the ratio between the length of the filter and signal is almost vanishing, though, making the edge issue negligible. But the applications in this thesis are of such nature that the edge problem needs attention. This is the reason for this and the following chapter.

The content of this chapter is a presentation of four different solutions to the edge problem. All are fairly simple and rely on basic linear algebra and calculus. The following chapter is dedicated entirely to a fifth solution, which requires some knowledge of the multiresolution analysis.

This chapter is a condensed and rewritten excerpt from Jensen and la Cour-Harbo [45].

9.1 Defining the Problem

There exists a number of different solutions to the edge problem. Common to those considered here is the preservation of the perfect reconstruction property of the wavelet transform. The three most often used ones is *edge filters*, *periodization*, and *mirroring*. An obvious and very simple, but unattractive solution called *zero padding* is presented first with the argumentation for not choosing this approach in any real applications.

9.2 Zero padding

The most obvious solution to the edge problem is to extend a finite signal to a infinite signal by applying zeros a both ends. This is called zero padding. In practice this means that when the computation of a coefficient in the transform requires a sample beyond the range of the given samples in the finite signal, the value zero is used.

Applying zero padding to a signal with 8 samples followed by the Haar transform yields (up to) 4 nonzero entries in each of the low and the high pass parts. Going through the steps in the Daubechies 4 transform will show that in the high pass part the entries

with indices $0, 1, 2, 3, 4$ can be nonzero, and in the low pass part those with indices $-1, 0, 1, 2, 3$ can be nonzero. Thus in the two components in the transform there may be a total of 10 nonzero samples. It is important to note that all 10 coefficients above are needed to reconstruct the original signal, so two of them cannot just be left out if the perfect reconstruction property is to be preserved. In general the number of extra coefficients is proportional to the filter length. For orthogonal transforms (such as those in the Daubechies family) the number of extra signal coefficients is exactly $L - 2$, with L being the filter length.

When using zero padding the growth in the number of nonzero entries is unavoidable. It is not a problem in the theory, but certainly in applications. Suppose a signal of length N is to be transformed with DWT over k scales with a filter of length L, where k is compatible with the length of the signal, i.e. $N \geq 2^k$. Each application of the DWT adds $L-2$ new nonzero coefficients, in general. Thus the final length of the transformed signal can be up to $N + k(L - 2)$.

The result of using zero padding is illustrated as in Fig. 9.1. As the filter taps "slides" across the signal a number of low and high pass transform coefficients are produced, a pair for each position of the filter. Since there are $(N + L)/2 - 1$ different positions, the total number of transform coefficients is twice this number, that is $N + L - 2$.

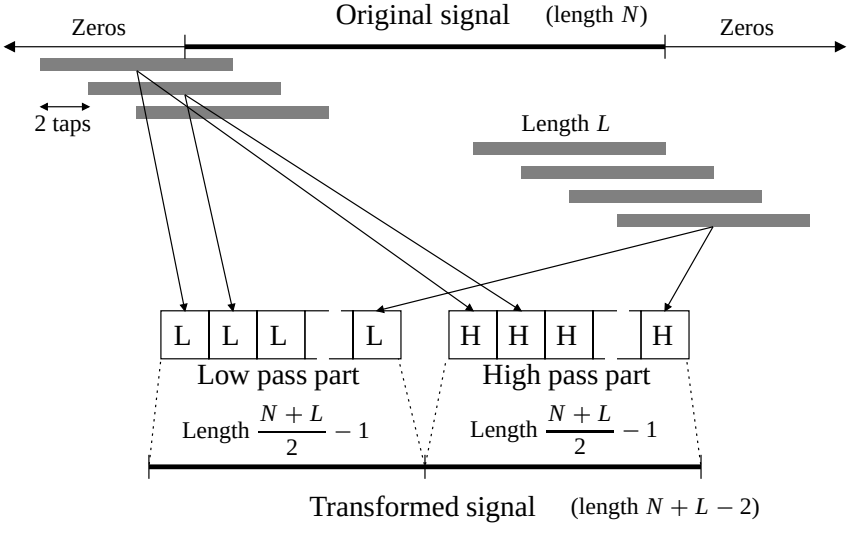


Figure 9.1: The result of zero padding when transforming a finite signal. The grey boxes illustrates the positions of the filter taps as the filtering occurs. Each position gives a low pass and high pass coefficient. The number of positions determines the number of transform coefficients. Note that most of the 'interior' filters have been left out to simplify this figure.

For wavelet packet decompositions the problem is much worse. Suppose one computes the full wavelet packet decomposition down to a level J, i.e. applying the DWT $J-1$ times, each time to all elements in the previous level. Starting with a signal of length N and a filter of length L, then at the level J the total length of the transformed signal can be up to $N + (2^{J-1} - 1)(L-2)$. This exponential growth in J makes zero padding an unattractive solution to the edge problem.

Thus it is preferable to have available edge correction methods, such that application of the corrected DWT to a signal leads to two components, each of half the length of the original signal. Furthermore preservation of the perfect reconstruction property is high desirable. Three different methods are presented below, and one in the following chapter. The first three methods use a number of results from linear algebra. The fourth method requires extensive knowledge of the classical wavelet theory and some harmonic analysis. Two of the solutions attempts to handle the problem by (initially) changing the signal, while two others introduces edge filters (a change of the transform).

The idea behind edge filters is to replace the filters in each end of the signal with some new filter coefficients designed to preserve both the length of the signal and the perfect reconstruction property. This idea is depicted in Fig. 9.2.

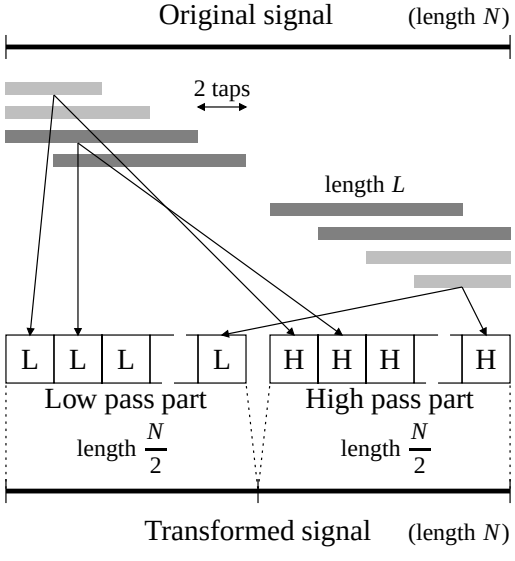


Figure 9.2: The idea behind all types of edge filters is to replace the filters reaching beyond the signal (see Fig. 9.1) with new, shorter filters (light grey). By having the right number of edge filters it is possible to get exactly the same number of transform coefficients as signal samples while preserving certain properties of the wavelet transform.

9.3 DWT as a Matrix

The probably most common interpretation of the DWT is as a low and high pass filtering followed by down sampling by 2. This is also the usual form of implementation of the transform. In this section the attention is turned to another possibility. Since the DWT is linear and (now assumed to be) finite, it can be carried out by multiplying the signal with an appropriate (non-singular) matrix. The reconstruction can of course also be done by a single multiplication. Note that in the following it is implicitly assumed that the input signal, denoted by $\mathbf{x}$, is of even length whenever finite.

Recall from (A.2) that the low pass filtered and down sampled signal is given as

$$(H\mathbf{x})[n] = \sum_k h[2n-k]x[k]\,. \tag{9.1}$$

This convolution is interpreted as an inner product between a zero padded $\mathbf{h}$ and $\mathbf{x}$, or as the matrix product of the reversed filter row vector and the signal column vector. The high pass part $G\mathbf{x}$ is found analogously, see (A.3). The symbols H and G emphasize that the transitions from $\mathbf{x}$ to $H\mathbf{x}$ and $G\mathbf{x}$ are linear maps. It is necessary to decide how to combine the coefficients of $H\mathbf{x}$ and $G\mathbf{x}$ into a single vector to get the matrix form of the transform. There are two obvious possibilities. One is to take all the components in $H\mathbf{x}$, followed by all components in $G\mathbf{x}$. This is not an easy solution to use, when one considers infinite signals. The other possibility is to interlace the components in a column vector as

$$\mathbf{y} = \begin{bmatrix}\cdots & (H\mathbf{x})[-1] & (G\mathbf{x})[-1] & (H\mathbf{x})[0] & (G\mathbf{x})[0] & (H\mathbf{x})[1] & (G\mathbf{x})[1] & \cdots\end{bmatrix}^\top.$$

Since the four vectors in an orthogonal filter set have equal even length (in contrast to most biorthogonal filter sets), it is easier to describe the matrix form of the DWT for orthogonal filters. Later on it is fairly easy to extend the matrix form to biorthogonal filters.

The rows of the matrix consist of alternating, reversed low and high pass IRs, each low pass IR is shifted two places in relations to the preceding high pass IR, while the following high pass IR is not shifted. The low pass filter is now denoted by $\mathbf{h}$ and the high pass filter by $\mathbf{g}$.

If the length of the filter is 6, then the matrix becomes

$$\mathbf{T}_{\mathrm{a}} = \begin{bmatrix}
\ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & & & & & \\
\ddots & h[5] & h[4] & h[3] & h[2] & h[1] & h[0] & 0 & 0 & 0 & 0 & \cdots \\
 & g[5] & g[4] & g[3] & g[2] & g[1] & g[0] & 0 & 0 & 0 & 0 & \cdots \\
\cdots & 0 & 0 & h[5] & h[4] & h[3] & h[2] & h[1] & h[0] & 0 & 0 & \cdots \\
\cdots & 0 & 0 & g[5] & g[4] & g[3] & g[2] & g[1] & g[0] & 0 & 0 & \cdots \\
\cdots & 0 & 0 & 0 & 0 & h[5] & h[4] & h[3] & h[2] & h[1] & h[0] & \\
\cdots & 0 & 0 & 0 & 0 & g[5] & g[4] & g[3] & g[2] & g[1] & g[0] & \ddots \\
 & & & & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots
\end{bmatrix}. \tag{9.2}$$

Given an infinite signal $\mathbf{x}$ as a column vector the wavelet transform can be calculated simply by $\mathbf{y} = \mathbf{T}_a\mathbf{x}$. Obviously the original signal should be reconstructable in the same manner, so another matrix is needed such that $\mathbf{T}_s\mathbf{T}_a = \mathbf{I}$. For finite matrices this implies that $\mathbf{T}_s = \mathbf{T}_a^{-1}$, and for infinite matrices this condition is imposed. Fortunately, it is easy to show that $\mathbf{T}_a^{-1} = \mathbf{T}_a^\top$ for orthogonal filters. Now $\mathbf{x} = \mathbf{T}_s\mathbf{y}$, so in order to reconstruct the original signal the matrix $\mathbf{T}_s$ is applied to a mix of low and high pass coefficients.

The major difference in the case of biorthogonal filters is that $\mathbf{T}_a$ is not orthogonal, and hence $\mathbf{T}_s$ cannot be found simply by transposing the direct transform matrix. To understand how $\mathbf{T}_s$ is constructed in this case, first examine $\mathbf{T}_s$ in the orthogonal case. It is easy to show that

$$\mathbf{T}_s = \mathbf{T}_a^\top = \begin{bmatrix}
\ddots & \ddots & & \vdots & \vdots & \vdots & \vdots & \\
\ddots & \tilde{h}[0] & \tilde{g}[0] & 0 & 0 & 0 & 0 & \\
\ddots & \tilde{h}[1] & \tilde{g}[1] & 0 & 0 & 0 & 0 & \\
\ddots & \tilde{h}[2] & \tilde{g}[2] & \tilde{h}[0] & \tilde{g}[0] & 0 & 0 & \\
\ddots & \tilde{h}[3] & \tilde{g}[3] & \tilde{h}[1] & \tilde{g}[1] & 0 & 0 & \\
\ddots & \tilde{h}[4] & \tilde{g}[4] & \tilde{h}[2] & \tilde{g}[2] & \tilde{h}[0] & \tilde{g}[0] & \\
 & \tilde{h}[5] & \tilde{g}[5] & \tilde{h}[3] & \tilde{g}[3] & \tilde{h}[1] & \tilde{g}[1] & \ddots \\
 & 0 & 0 & \tilde{h}[4] & \tilde{g}[4] & \tilde{h}[2] & \tilde{g}[2] & \ddots \\
 & 0 & 0 & \tilde{h}[5] & \tilde{g}[5] & \tilde{h}[3] & \tilde{g}[3] & \ddots \\
 & 0 & 0 & 0 & 0 & \tilde{h}[4] & \tilde{g}[4] & \ddots \\
 & 0 & 0 & 0 & 0 & \tilde{h}[5] & \tilde{g}[5] & \ddots \\
 & \vdots & \vdots & \vdots & \vdots & & \ddots & \ddots
\end{bmatrix} \tag{9.3}$$

for a length 6 orthogonal filter.

In the same way $\mathbf{T}_s$ can be written for biorthogonal filters, except with the obvious difference that there is not the close connection between analysis and synthesis that characterized the orthogonal filters.

The matrices of the direct and inverse transforms have been introduced in order to explain how to construct edge filters. Computationally both filtering and lifting are much more efficient transform implementations.

9.4 Gram-Schmidt Edge Filters

The construction of edge filters begins by looking more carefully at the problem with zero padding. Suppose a finite signal $\mathbf{x}$ of length N is given. First zero padding is applied, creating the new signal $\mathbf{s}$ of infinite length, by defining

$$s[n] = \begin{cases} 0 & \text{if } n \le -1\ , \\ x[n] & \text{if } n = 0, 1, \ldots, N-1\ , \\ 0 & \text{if } n \ge N\ . \end{cases} \tag{9.4}$$

Suppose that the filter has length L, with the nonzero coefficients having indices between 0 and $L-1$. To avoid special cases assume also that N is substantially larger than L, and that both L and N are even. Examine now (see (A.2))

$$(H\mathbf{s})[n] = \sum_{k\in\mathbb{Z}} h[2n-k]s[k] = \sum_{k=0}^{N-1} h[2n-k]x[k]$$

for each possible value of n. If $n < 0$, the sum is always zero. The first nonzero term can occur when $n = 0$, in which case $(H\mathbf{s})[0] = h[0]x[0]$. The last nonzero term occurs for $n = (N+L-2)/2$, and it is $(H\mathbf{s})[(N+L-2)/2] = h[L-1]x[N-1]$. The same computation is valid for the $G\mathbf{s}$ vector. Thus in the transformed signal the total number of nonzero terms can be up to $N+L-2$.

This computation also shows that in the index range $L/2 < n < N-(L/2)$ all filter coefficients are multiplied with **x**-entries. At the start and the end only some filter coefficients are needed, the others being multiplied by zero from the zero padding of the signal **s**. This leads to the introduction of the edge filters. The filters are modified during the $L/2$ evaluations at both the beginning and the end of the signal, taking into account only those filter coefficients that are actually needed. Thus to adjust the **h** filter a total of L new filters will be needed. The same number of modifications will be needed for the high pass filter. It turns out that fewer modified filters are needed, if the location of the finite signal is shifted one unit.

So repeat now the computation above with the following modification of the zero padding. Define

$$s_{\text{shift}}[n] = \begin{cases} 0 & \text{if } n \le -2\,, \\ x[n+1] & \text{if } n = -1, 0, 1, \ldots, N-2\,, \\ 0 & \text{if } n \ge N-1\,. \end{cases} \tag{9.5}$$

With this modification the first non-zero term in $H\mathbf{s}$ can be

$$(H\mathbf{s}_{\text{shift}})[0] = h[1]x[0] + h[0]x[1]\,,$$

and the last nonzero term can be

$$(H\mathbf{s}_{\text{shift}})[(N+L)/2-2] = h[L-1]x[N-2] + h[L-2]x[N-1]\,,$$

due to the assumption that L is even. With this shift a total of $L-2$ corrections are needed at each end. This shifted placement of the non-zero coefficients will be used in the next subsection.

9.4.1 The DWT Matrix Applied to Finite Signals

Instead of using zero padding the matrices $\mathbf{T}_a$ and $\mathbf{T}_s$ could be truncated, by removing the parts multiplying the zero padded parts of the signal. Although this gives finite matrices

it does not solve the problem that the transformed signal can have more nonzero entries than the original signal. The next step is therefore to alter the truncated matrices to get orthogonal matrices (only orthogonal filters will be treated here).

The procedure is first discussed using an example with a relatively short filter (otherwise the matrices will be rather big). A generalization is presented in Section 9.4.2. For a filter of length 6 and a signal of length 8 the transformed signal can have 12 non-vanishing elements, as was shown previously. The the part of the matrix that multiplies zeros in $\mathbf{s}_{\text{shift}}$ is now removed, and the result, denoted by $\mathbf{T}'_{\text{a}}$, is given as

$$\mathbf{T}'_{\text{a}}\mathbf{x} = \begin{bmatrix} h[1] & h[0] & 0 & 0 & 0 & 0 & 0 & 0 \\ g[1] & g[0] & 0 & 0 & 0 & 0 & 0 & 0 \\ h[3] & h[2] & h[1] & h[0] & 0 & 0 & 0 & 0 \\ g[3] & g[2] & g[1] & g[0] & 0 & 0 & 0 & 0 \\ h[5] & h[4] & h[3] & h[2] & h[1] & h[0] & 0 & 0 \\ g[5] & g[4] & g[3] & g[2] & g[1] & g[0] & 0 & 0 \\ 0 & 0 & h[5] & h[4] & h[3] & h[2] & h[1] & h[0] \\ 0 & 0 & g[5] & g[4] & g[3] & g[2] & g[1] & g[0] \\ 0 & 0 & 0 & 0 & h[5] & h[4] & h[3] & h[2] \\ 0 & 0 & 0 & 0 & g[5] & g[4] & g[3] & g[2] \\ 0 & 0 & 0 & 0 & 0 & 0 & h[5] & h[4] \\ 0 & 0 & 0 & 0 & 0 & 0 & g[5] & g[4] \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \\ x[5] \\ x[6] \\ x[7] \end{bmatrix} = \begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ y[3] \\ y[4] \\ y[5] \\ y[6] \\ y[7] \\ y[8] \\ y[9] \\ y[10] \\ y[11] \end{bmatrix}. \tag{9.6}$$

It is evident from the two computations above with the original and the shifted signal that the truncation of the $\mathbf{T}_{\text{a}}$ matrix is not unique. As described above, choice here is to align the first non-vanishing element in $\mathbf{x}$ with $h[1]$ and $g[1]$. This makes $\mathbf{T}'_{\text{a}}$ "more symmetric" than if $h[0]$ and $g[0]$ had been chosen. Moreover, choosing the symmetric truncation guarantees linear independence of the rows, see [40], a property which will be needed later. Applying the same type of truncation the synthesis matrix, i.e. reducing $\mathbf{T}_{\text{s}}$ to an 8×12 matrix $\mathbf{T}'_{\text{s}}$, yields $\mathbf{T}'_{\text{s}}\mathbf{T}'_{\text{a}} = \mathbf{I}$, so perfect reconstruction is still possible.

The next step is to change $\mathbf{T}'_{\text{a}}$ such that $\mathbf{y}$ has the same number of coefficients as $\mathbf{x}$. When looking at the matrix equation (9.6) the first idea might be to further reduce the size of $\mathbf{T}'_{\text{a}}$, this time making an 8×8 matrix by removing the two upper and lower most rows. That is

$$\mathbf{T}''_{\text{a}} = \begin{bmatrix} h[3] & h[2] & h[1] & h[0] & 0 & 0 & 0 & 0 \\ g[3] & g[2] & g[1] & g[0] & 0 & 0 & 0 & 0 \\ h[5] & h[4] & h[3] & h[2] & h[1] & h_0 & 0 & 0 \\ g[5] & g[4] & g[3] & g[2] & g[1] & g_0 & 0 & 0 \\ 0 & 0 & h[5] & h[4] & h[3] & h[2] & h_1 & h_0 \\ 0 & 0 & g[5] & g[4] & g[3] & g[2] & g_1 & g_0 \\ 0 & 0 & 0 & 0 & h[5] & h[4] & h[3] & h_2 \\ 0 & 0 & 0 & 0 & g[5] & g[4] & g[3] & g_2 \end{bmatrix} \tag{9.7}$$

At least this will ensure a transformed signal with only 8 coefficients. Removing the two first and two last columns in $\mathbf{T}'_{\text{s}}$ produces an 8×8 synthesis matrix. These matrices

cannot fulfill the perfect reconstruction condition $\mathbf{T}_{\mathrm{s}}''\mathbf{T}_{\mathrm{a}}'' = \mathbf{I}$; the 1's on the diagonal of $\mathbf{T}_{\mathrm{s}}'\mathbf{T}_{\mathrm{a}}'$ comes from $\sum_{n=0}^{5} |h_n|^2 = 1$, and since the diagonal of $\mathbf{T}_{\mathrm{s}}''\mathbf{T}_{\mathrm{a}}''$ contains partial sums of this sum, the diagonal cannot be all 1's. But $\mathbf{T}_{\mathrm{a}}''$ and $\mathbf{T}_{\mathrm{s}}''$ both have full rank. This is made plausible by the following argument. Suppose the first and second row of $\mathbf{T}_{\mathrm{a}}''$ are linearly dependent, i.e.

$$\alpha \begin{bmatrix} h_3 & h_2 & h_1 & h_0 \end{bmatrix} = \begin{bmatrix} h_2 & -h_3 & h_4 & -h_5 \end{bmatrix},$$

then $\alpha h_3 = h_2$ and $\alpha h_2 = -h_3$ implying that $\alpha = \pm i$, which is clearly inadmissible. Consequently, the orthogonality can be restored by using the Gram-Schmidt orthogonalization procedure.

9.4.2 The General Case

In the previous section the construction of edge filters was partly demonstrated using a particular filter, but it is not difficult to generalize the method. For any wavelet filter it is always possible to truncate the corresponding analysis matrix $\mathbf{T}_{\mathrm{a}}$, such that the result is an $N \times N$ matrix $\mathbf{M}$ (with N even) with all but the first and last $L/2-1$ rows containing whole IRs, and such that the upper and lower truncated rows have an equal number of non-vanishing entries. If $L = 4K+2$, $K \in \mathbb{N}$, the first row in $\mathbf{M}$ will be (a part of) the low pass IR $\mathbf{h}$, and if $L = 4K$ the first row will be (a part of) the high pass IR $\mathbf{g}$. It can be shown (see [39]) that this symmetric truncation always produces a full rank matrix. All the truncated rows are orthogonalized by the Gram-Schmidt procedure. Since the rows containing whole IRs are mutual orthogonal, and since all rows containing truncated IRs are orthogonal to all of the rows containing whole IRs, the first $L/2-1$ rows need to be orthogonalized (with respect to themselves). The same applies to the last $L/2-1$ rows. So the left edge filters $\mathbf{m}_k^{\mathrm{l}}$ are defined as

$$\mathbf{m}_k' = \mathbf{m}_k - \sum_{n=0}^{k-1} \frac{\mathbf{m}_n \mathbf{m}_k^{\top}}{\|\mathbf{m}_n\|^2} \mathbf{m}_n, \qquad \mathbf{m}_k^{\mathrm{l}} = \frac{\mathbf{m}_k'}{\|\mathbf{m}_k'\|^2}, \qquad k = 0, 1, \ldots, L/2-2\,.$$

Note that this order of orthogonalization preserves the staggered length (i.e. number of non-vanishing coefficients) of the left edge filters. In the same way the vectors $\mathbf{m}_{N-L/2+2}$ through $\mathbf{m}_{N-1}$ are converted into $L/2-1$ right edge filters, which is denoted $\mathbf{m}_0^{\mathrm{r}}$ through $\mathbf{m}_{L/2-1}^{\mathrm{r}}$. The Gram-Schmidt orthogonalization of the right edge filters starts with $\mathbf{m}_{N-1}$.

The new orthogonal matrix then becomes

$$\mathbf{M}' = \begin{bmatrix} \mathbf{m}_0^{\mathrm{l}} \\ \vdots \\ \mathbf{m}_{L/2-2}^{\mathrm{l}} \\ \mathbf{m}_{L/2-1} \\ \vdots \\ \mathbf{m}_{N-L/2+2} \\ \mathbf{m}_0^{\mathrm{r}} \\ \vdots \\ \mathbf{m}_{L/2-2}^{\mathrm{r}} \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ b \\ c \end{matrix}}\right\} L/2-1 \text{ left edge filters}, \\ \left.\vphantom{\begin{matrix} a \\ b \\ c \end{matrix}}\right\} N-L+2 \text{ whole filters}, \\ \left.\vphantom{\begin{matrix} a \\ b \\ c \end{matrix}}\right\} L/2-1 \text{ right edge filters}, \end{matrix} \tag{9.8}$$

The edge filters belonging to the inverse transform are easily found, since the synthesis matrix is the transpose of analysis matrix.

9.5 Periodization

The simple solution to the edge problem was zero padding. Another possibility is to choose samples from the signal to use for the missing samples. One way of doing this is to *periodize* the finite signal. Suppose the original finite signal is the column vector $\mathbf{x}$, of length N. Then the periodized signal $\mathbf{x}^{\mathrm{p}}$ is given as a vertical concatenation of infinitely many $\mathbf{x}$. This signal is periodic with period N, since $x^{\mathrm{p}}[k+N] = x^{\mathrm{p}}[k]$ for all $k \in \mathbb{Z}$. It is important to note that the signal $\mathbf{x}^{\mathrm{p}}$ has infinite energy. But it can still be transformed with $\mathbf{T}_{\mathrm{a}}$, since the filters are of finite length, such that each row in $\mathbf{T}_{\mathrm{a}}$ only has a finite number of nonzero entries. Since $\mathbf{y}^{\mathrm{p}} = \mathbf{T}_{\mathrm{a}}\mathbf{x}^{\mathrm{p}}$ is periodic with period N, this formula defines a finite signal $\mathbf{y}$ by selecting N consecutive samples from $\mathbf{y}^{\mathrm{p}}$. Note that the choice of these entries is not unique. The same procedure can be used to inversely transform $\mathbf{y}$ into $\mathbf{x}$ using the infinite $\mathbf{T}_{\mathrm{s}}$. Thus periodization is a way of transforming a finite signal while preserving the length of it. In implementations only enough samples to cover the extent of the filters are needed, which is at most $L-2$. It is of course desirable to avoid extending the signal at all, since this requires extra time and memory in an implementation. Fortunately, it is very easy to alter the transform matrix to accommodate this desire.

First the infinite transform matrix is truncated such that it fits the signal, i.e. for a signal of length N, the matrix is reduced to an $N \times N$ matrix. The matrix now consists of some whole IRs and some truncated IRs. The removed filter taps from the latter is then inserted in the matrix N positions to the right (in the upper part of the matrix) or to the left (in the lower part) of their locations prior to truncation.

This is easily visualized with an example. For a signal of length 10 and filter of length

6 the truncated matrix performing transformation by periodization is given as

$$\mathbf{T}_{\mathrm{a}}^{\mathrm{p}} = \begin{bmatrix} h[3] & h[2] & h[1] & h[0] & 0 & 0 & 0 & 0 & h[5] & h[4] \\ g[3] & g[2] & g[1] & g[0] & 0 & 0 & 0 & 0 & g[5] & g[4] \\ h[5] & h[4] & h[3] & h[2] & h[1] & h[0] & 0 & 0 & 0 & 0 \\ g[5] & g[4] & g[3] & g[2] & g[1] & g[0] & 0 & 0 & 0 & 0 \\ 0 & 0 & h[5] & h[4] & h[3] & h[2] & h[1] & h[0] & 0 & 0 \\ 0 & 0 & g[5] & g[4] & g[3] & g[2] & g[1] & g[0] & 0 & 0 \\ 0 & 0 & 0 & 0 & h[5] & h[4] & h[3] & h[2] & h[1] & h[0] \\ 0 & 0 & 0 & 0 & g[5] & g[4] & g[3] & g[2] & g[1] & g[0] \\ h[1] & h[0] & 0 & 0 & 0 & 0 & h[5] & h[4] & h[3] & h[2] \\ g[1] & g[0] & 0 & 0 & 0 & 0 & g[5] & g[4] & g[3] & g[2] \end{bmatrix}. \tag{9.9}$$

Then $\mathbf{y} = \mathbf{T}_{\mathrm{a}}^{\mathrm{p}}\mathbf{x}$ is equal to one particular choice of N consecutive samples in $\mathbf{y}^{\mathrm{p}}$. Now $\mathbf{T}_{\mathrm{a}}^{\mathrm{p}}$ is orthogonal, so the inverse transform is given by $(\mathbf{T}_{\mathrm{a}}^{\mathrm{p}})^{\top}$. Note that the symmetric structure of the matrix is not a necessity for the preservation of length and energy.

The same principle can be applied to biorthogonal filters. For an example of this, see Jensen and la Cour-Harbo [45].

Moment Preserving Edge Filters

10

The methods for handling the edge problem presented so far have focused on maintaining the orthogonality of the transform. Orthogonality is important, since it is equivalent with energy preservation. But there are other properties beside energy which can prove useful to preserve under transformation. One of them is related to moments of a sequence, which in turn is related to the ability of the wavelets to approximate functions in C^r, that is spaces of r times continuous-differentiable functions, disregarding that the wavelets do not form an orthogonal basis for C^r (they form unconditional bases).

This chapter is divided into a number of sections, starting with an introduction to moments of sequences and why the preservation of moments is relevant. Then in Section 10.3 the edge scaling functions and wavelets are derived. This is done mostly in the form of proofs of lemmas and theorems. It turns out that an extra step is needed to complete the construction. This is presented and discussed in Section 10.4. Two examples of edge functions and filters are then given in Section 10.5, and the numerical stability of the procedure is discussed in Section 10.6. Finally, the chapter ends with conclusion in Section 10.8.

The majority of the results in this chapter is from Cohen et al. [22].

10.1 The Idea of Moment Preservation

This first section is dedicated to a brief and incomplete explanation of the idea behind the construction of moment preserving edge filters. The incitement for being concerned with the subject is given in the following section, while a rather short 'recipe' is provided in Section 10.1.2.

10.1.1 Why Moment Preserving Transforms?

The answer to this question begins by making some observations on the smoothness of the wavelet ψ. For all wavelets it is true that $\psi \in C^M(\mathbb{R})$ for some $M \in \mathbb{N}$ (see for instance Corollary 5.5.2 in Daubechies [26, p. 154]), which means that (see Cohen et al. [22])

$$\int t^k \psi(t) dt = \frac{d^k}{d\xi^k} m_0(\xi)\Big|_{\xi=\pi} = 0, \quad k = 0, \ldots, M-1,$$

and this is equivalent to

$$\sum_n n^k g_n = 0, \quad k = 0, \ldots, M-1, \tag{10.1}$$

where $\mathbf{g}$ is the corresponding IR. Note that this implies that $\mathbf{g}$ has at least $2M$ non-vanishing coefficients. For Daubechies $2N$ this property holds for $M = N$, since the Daubechies $2N$ wavelets are constructed as

$$m_0(\xi) = \left(\frac{1+e^{-i\xi}}{2}\right)^N Q_N(\xi), \tag{10.2}$$

where Q_N is a polynomial of order $N-1$ in $e^{-i\xi}$. For CDF(2,2) $M = 2$, and for CDF(4,6) $M = 4$. A sequence satisfying (10.1) for some M is said to have M *vanishing moments*.

Assume that the filter $\mathbf{g}$ has M vanishing moments. Take a polynomial $p(t) = \sum_{j=0}^{M-1} p_j t^j$ of degree at most $M-1$. Take a signal obtained by sampling this polynomial at the integers, i.e. $s[n] = p(n)$, and filter this signal with $\mathbf{g}$.

$$\begin{aligned}
(\mathbf{g} * \mathbf{s})_n &= \sum_k g_k s_{n-k} \\
&= \sum_k g_k \sum_{j=0}^{M-1} p_j (n-k)^j \\
&= \sum_k g_k \sum_{j=0}^{M-1} p_j \sum_{m=0}^{j} \binom{j}{m} (-1)^m k^m n^{j-m} \\
&= \sum_{j=0}^{M-1} p_j \sum_{m=0}^{j} \binom{j}{m} (-1)^m n^{j-m} \sum_k k^m g_k = 0\,.
\end{aligned} \tag{10.3}$$

Note that $\mathbf{g}$ is of finite length, so all sums above are finite. Thus filtering with $\mathbf{g}$ maps a signal obtained from sampling a polynomial of degree at most $M-1$ to zero. Note also that the polynomial need not be sampled at the integers. It is sufficient that the sample points are equidistant.

This property of the high pass filter $\mathbf{g}$ has an interesting consequence when applying a DWT. Because of the perfect reconstruction property, the polynomial samples get mapped into the low pass part. This is consistent with the intuitive notion that polynomials of low degree do not oscillate much, meaning that they contain no high frequencies. The computation in (10.3) shows that with the particular filters used here (that is, with sufficiently many vanishing moments) the high pass part is actually zero, and not just close to zero, as is typical for non-ideal filters (at this point one should recall the filters used in a wavelet decomposition are not ideal).

Due to these properties it would be interesting to have an edge correction method which preserved vanishing moments of finite signals of a given length. Such a method

was found by A. Cohen, I. Daubechies, and P. Vial [22]. Their solution is presented in detail in the following sections.

To illustrate what moment preserving filters can accomplish, Fig. 10.3 shows the result of using three different edge handling methods.

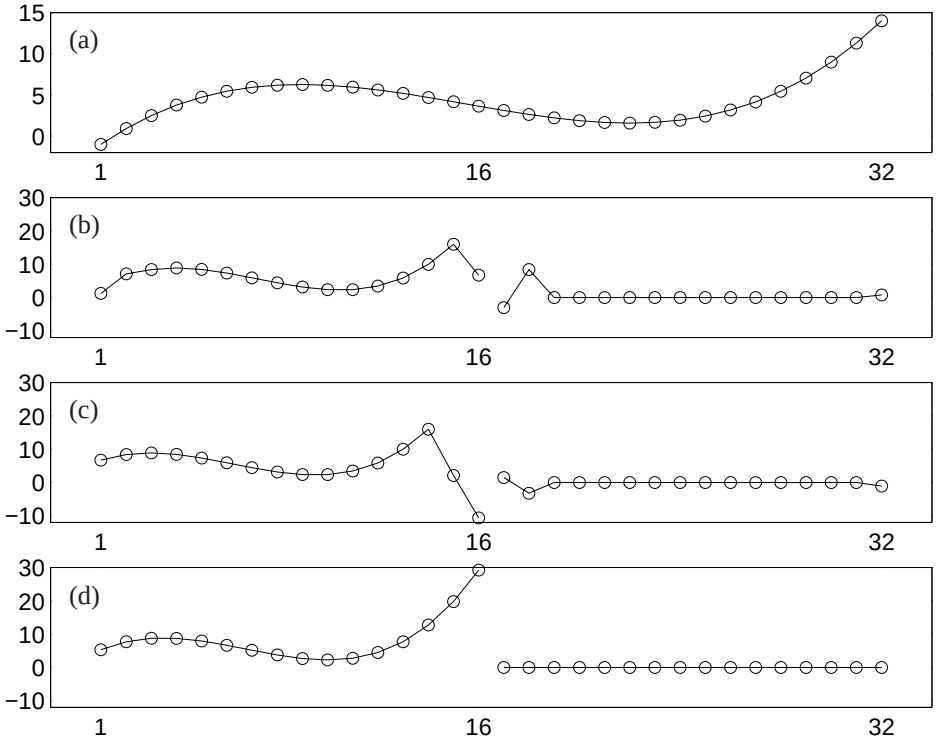


Figure 10.1: In (a) the polynomial $t^3 - 4t^2 + 2t + 6$ is sampled in 32 points in the interval $[-1; 4]$. The following graphs (b), (c), and (d) show the result of transforming using periodization, Gram-Schmidt orthogonalized edge filters, and moment preserving edge filters. Daubechies length 8 filters are used. The (d) graph clearly shows the advantage of moment preserving filters, as the high pass part is completely vanishing (as it should be according to (10.3)), while the low pass part is another third degree polynomial sampled in 16 points.

10.1.2 How to Make a Moment Preserving Transform

The idea for preserving the number of vanishing moments, and hence be able to reproduce polynomials completely in the low pass part, is simple, although the computations are non-trivial. This section begins with a brief review of the reason for and idea behind the transform which can reproduce polynomials.

Redoing the computation in (10.3) for a general filter $\mathbf{h}$ of length L yields

$$\begin{aligned}(\mathbf{h} * \mathbf{s})_n &= \sum_{k=0}^{L-1} h_k s_{n-k} = \sum_{k=0}^{L-1} h_k \sum_{j=0}^{M-1} p_j (n-k)^j \\ &= \sum_{k=0}^{L-1} h_k \sum_{j=0}^{M-1} p_j \sum_{m=0}^{j} \binom{j}{m} n^m (-k)^{j-m} = \sum_{m=0}^{M-1} q_m n^m \ , \end{aligned} \tag{10.4}$$

where complicated expressions for the coefficients q_m are omitted, since they are not needed. This computation shows that convolution with any filter $\mathbf{h}$ of finite length takes a sampled polynomial of degree at most $M-1$ into another sampled polynomial, again of degree at most $M-1$. If the signal in question have a finite number of nonzero samples, then the resulting convolution will have more samples, as explained above.

This computation must be invertible. This means that if the starting point is the signal $\mathbf{x}$ of length N, obtained by sampling an arbitrary polynomial of degree at most $M-1$,

$$x[n] = \sum_{m=0}^{M-1} q_m n^m, \quad n = 0, \ldots, N-1 \ ,$$

then the goal is to find another polynomial p of degree at most $M-1$, and a signal $\mathbf{s}$ of the same length N, obtained by sampling p, such that $\mathbf{x} = \mathbf{h} * \mathbf{s}$. To do this for all polynomials of degree at most $M-1$ yields a set of edge filters, in a way similar to the constructions already done in Section 9.4.

As described previously the idea is to make corrections to the filters used at the start and end of the signal in order to preserve vanishing moments for signals of a fixed finite length. It is done as follows. The first (leftmost) edge filter on the left and the last (rightmost) edge filter on the right is chosen such that they preserve vanishing of the moment of order $m = 0$ in the high pass part. The next pair is chosen such that moments or order $m = 1$ vanish. This continues until M edge filters are produced for the transform under consideration.

It is by no means trivial to construct the edge filters and to prove that the described procedure does produce a moment preserving transform, and it takes further computations to make these new edge filters both orthogonal and of decreasing length.

Unfortunately, the efforts so far are not enough to construct a transform applicable to finite signals, such that it preserves vanishing moments. Thus the description so far is incomplete. Briefly, what remains to be done is an extra step, which consists in pre-conditioning the signal prior to transformation by multiplying the first M and the last M

samples by an $M \times M$ matrix. After transformation the result is multiplied by the inverse of this matrix, at the beginning and end of the signal.

10.2 Polynomials and Wavelet Bases

Before venturing into the comprehensive description of the construction of moment preserving edge scaling functions it is necessary to understand in more detail the relation between polynomials of a certain order and wavelets with an equal number of vanishing moments.

10.2.1 Polynomials on the Real Line

The first thing to do is to establish that the scaling functions actually do generate polynomials up to the degree of the number of vanishing moments, and in what sense this is valid. To investigate this, first define the set in question. Let

$$\mathcal{P}_N(I) \equiv \left\{ f(t) \;\middle|\; f(t) = \sum_{n=0}^{N-1} a_n t^n,\ t \in I \subseteq \mathbb{R}, a_n \in \mathbb{R} \right\}.$$

Note that $\mathcal{P}_N(I)$ is vector space.

As described in the previous section we want to do approximations in some instance of this space (for a certain choice of I), preferably for $I = \mathbb{R}$. The approximation should be by translated version of the scaling functions (no dilation) because the vanishing moments of $\psi(t)$ leads us to believe that one level of scaling functions is enough to span polynomials with sufficiently low degree. This expectation is expressed by the mapping of sampled polynomials to the zero sequence by the high pass filter in (10.3). Thus, we are interested in an approximation on the form

$$p(t) = \sum_{n \in \mathbb{Z}} e_n \phi(t - n), \qquad p \in \mathcal{P}_N(I).$$

There are significant difference between $\mathcal{P}_N(\mathbb{R})$ and $\mathcal{P}_N(I)$, where I is a compact set. The latter is a space with much more structure, in particular, it is a Hilbert space in the L^2 inner product, as will be demonstrated shortly. But first a general statement on the approximation is given. This theorem, or rather the proof of it, contains much of the work in linking polynomials and wavelets.

Theorem 10.1 (Generation of Polynomials)
Let $\phi(t)$ be a scaling function with $m_0^{(k)}(\pi) = 0$ for $k = 0, \ldots, N-1$. Then for any $p \in \mathcal{P}_N(\mathbb{R})$ there exists a sequence $e_n \in \mathbb{R}$ such that

$$\sum_{n=-M}^{M} e_n \phi(t - n) \to p(t) \tag{10.5}$$

pointwise and locally uniformly on $\mathbb{R}$ *for* $M \to \infty$.

Local uniform convergence on $\mathbb{R}$ means uniform convergence on any bounded subset of $\mathbb{R}$.

The following proof is from Cohen et al. [22].

Proof

First note that

$$\sum_n (t-n)^k \phi(t-n) = C_k \ , \tag{10.6}$$

where C_k is a constant for fixed k. This is seen in the following way. Since (10.6) is periodic with period 1 it is completely characterized by its Fourier coefficients

$$\begin{aligned}\int_0^1 \sum_n (t-n)^k \phi(t-n) e^{-i2\pi rt} dt &= \sum_n \int_0^1 (t-n)^k \phi(t-n) e^{-i2\pi r(t-n)} dt \\ &= \int_{\mathbb{R}} t^k \phi(t) e^{-i2\pi rt} dt = \sqrt{2\pi}\big[t^k \phi(t)\big]^\wedge(2\pi r) = i^k \sqrt{2\pi}\hat{\phi}^{(k)}(2\pi r) \ .\end{aligned}$$

The last equality is a property of the Fourier transform, and can be formally derived as

$$i^k \hat{f}^{(k)}(t) = \frac{i^k}{\sqrt{2\pi}} \frac{d^k}{dt^k} \int_{\mathbb{R}} f(\xi) e^{-i\xi t} dt = \frac{i^k}{\sqrt{2\pi}} \int_{\mathbb{R}} f(\xi)(-i\xi)^k e^{-i\xi t} dt = \big[t^k f(t)\big]^\wedge(\xi) \ .$$

Since

$$\hat{\phi}^{(k)}(2\pi r) = \frac{d^k}{d\xi^k}\big[m_0(\xi)\hat{\phi}(\xi)\big]\Big|_{\xi=\pi r} = 2^{-k} \sum_{n=0}^{k} \binom{k}{n} m_0^{(k)}(\pi r) \hat{\phi}^{(k-n)}(\pi r) \ , \tag{10.7}$$

and since $m_0(\xi)$ is 2π periodic and $m_0^{(s)}(\pi) = 0$ for $s = 0, \ldots, N-1$, it follows that $\hat{\phi}^{(k)}(2\pi r) = 0$ for r odd, and by applying (10.7) to $\hat{\phi}^{(k-n)}(\pi r)$ that $\hat{\phi}^{(k)}(2\pi r) = 0$ for $r \neq 0$ even. This implies that (10.6) is constant. Moreover,

$$\begin{aligned} C_k &= \sqrt{2\pi}\big[t^k \phi(t)\big]^\wedge(0) \\ &= 2^{-1/2} \sum_m h_m \int_{\mathbb{R}} t^k \phi(2t-m) dt \\ &= 2^{-k-3/2} \sum_m h_m \int_{\mathbb{R}} (y+m)^k \phi(y) dy \\ &= 2^{-k-3/2} \sum_m h_m \int_{\mathbb{R}} \sum_{n=0}^{k} \binom{k}{n} m^n y^{k-n} \phi(y) dy \\ &= 2^{-k-1} \sum_{n=0}^{k} \binom{k}{n} 2^{-1/2} \sum_m h_m m^n \int_{\mathbb{R}} y^{k-n} \phi(y) dy \end{aligned}$$

$$= 2^{-k-1} \sum_{n=0}^{k} \binom{k}{n} M_n C_{k-n} \ ,$$

where the last equality follows from

$$C_k = \int_0^1 C_k dt = \int_0^1 \sum_n (t-n)^k \phi(t-n) dt = \int_{\mathbb{R}} t^k \phi(t) dt$$

and from defining $M_n = 2^{-1/2} \sum_m h_m m^n$. In order to prove (10.5) it suffices to show that

$$t^k = \sum_n e_{n,k} \phi(t-n), \quad k = 0, \ldots, N-1. \tag{10.8}$$

As a consequence of (10.6)

$$\begin{aligned} \sum_n n^k \phi(t-n) &= \sum_n (t-(t-n))^k \phi(t-n) \\ &= \sum_n \sum_{m=0}^{k} \binom{k}{m} t^{k-m} (-1)^m (t-n)^m \phi(t-n) \\ &= \sum_{m=0}^{k} \binom{k}{m} (-1)^m t^{k-m} C_m \\ &= t^k + \sum_{m=1}^{k} \binom{k}{m} (-1)^m t^{k-m} C_m. \end{aligned} \tag{10.9}$$

This means that t^k for $k = 0, \ldots, N-1$ can be written as a linear combination of ϕ and t^n for $n = 0, \ldots, k-1$, thus proving (10.8).

The convergence in (10.5) is only pointwise and not uniformly on $\mathbb{R}$ since we cannot have for any $\epsilon > 0$ that $|\sum_{-M}^{M} e_n \phi(t-n) - p(t)| < \epsilon$ for any M. This is because ϕ has compact support and not all function of $\mathcal{P}_N(\mathbb{R})$ converges to zero in $\pm\infty$. However, we do have local uniform convergence, that is uniform convergence on any bounded subset of $\mathbb{R}$, since all the involved functions are continues on $\mathbb{R}$. □

We would like to describe the relation between wavelets and polynomials in the setting of a more structured space. Wavelets and scaling functions are in the context of this thesis always used as building blocks for orthonormal bases, and it is therefore natural to look for a space with enough structure to introduce an orthonormal basis. Thus an inner product is needed. Since the usual L^2 inner product is obviously not an inner product in $\mathcal{P}_N(\mathbb{R})$, some other definition is needed. For instance

$$\langle p_1, p_2 \rangle_{\mathcal{P}_N} \equiv \int_{\mathbb{R}} (1+|t|^2)^{-N} p_1(t) p_2(t) dt \ . \tag{10.10}$$

Since $\mathcal{P}_N(\mathbb{R})$ is complete with the induced norm (it is isomorphic to $\mathbb{R}^N$), this makes $\mathcal{P}_N(\mathbb{R})$ a Hilbert space. The $\phi(t-n)$ is indeed a basis in the induced norm, but it is not an orthogonal basis, nor do the $\phi(t-n)$ have unit norm. In fact, the norm is rapidly decreasing due to the shift variant property of the inner product (10.10). This will cause the basis coefficients to blow up when approximating polynomials. Alternatively, orthonormalization of $\phi(t-n)$ (and thus bringing the basis coefficients into ℓ^2) will produce a set of functions with rapidly increasing sup-norm.

The blow up of either coefficients or the magnitude of the basis function demonstrates the difficulties in $\mathcal{P}_N(\mathbb{R})$. Another inner product will not solve this problem, because of the fundamental and intrinsic difference between polynomials on $\mathbb{R}$ and bases for $L^2(\mathbb{R})$.

One of the properties shared by polynomials and wavelets with a sufficient number of vanishing moments is differentiability. Thus, it seems obvious to investigate the relation in a Hölder space setting. It is true (see Meyer [58]) that if $\psi \in C^r(\mathbb{R})$ then $\phi(t-n)$ and $\psi_{-j,n}$, $j \in \mathbb{N}$ and $n \in \mathbb{Z}$, provide a unconditional basis (the convergence of the approximation is dependent only on the magnitude of the coefficients) for the function space C^s, for all $s < r$. Since C^s is a Banach space a norm (the Hölder norm) is available. Unfortunately, the polynomials defined on $\mathbb{R}$ are not in any Hölder space, as such a space contains only $L^\infty(\mathbb{R})$ functions. If this constraint is disregarded the resulting space is a Fréchet space, which is a complete metric space. But we do not have a Banach space since no single norm exists which induces a valid metric on the space. See Trèves [78].

10.2.2 Polynomials on the Interval

The above discussion indicates that as long as we confine the polynomials to a bounded interval on the real axis everything works out fine. This is true in the sense that $\mathcal{P}_N(I)$ is a complete subspace of $L^2(I)$, and (10.5) converges uniformly on I. Moreover, $\phi(t-n)$ is a basis for this space. It is not an orthogonal basis, however. And the $\phi(t-n)$ with support partially inside, and partially outside I, the so-called edge scaling functions, obviously do not integrate in square to 1 on I. A simple scaling will solve the latter problem in the sense that the internal ϕ together with the re-scaled edge scaling functions is a normed basis for $\mathcal{P}_N(I)$ (this approach, first suggested by Meyer, is discussed in [22, Sect. 3]). Unfortunately, this also introduces potentially severe numerical instability. The scaling functions with the main part of their L^2 norm outside I can require an arbitrarily large scaling to achieve unit norm when restricted to the interval. This is especially so for the scaling functions where the tails tends to zero so fast that the support, by visual inspection, seems somewhat smaller than it really is. For instance the scaling function in Fig. 10.2(d) seems to have support $[1; 6]$ where it actually is $[0; 9]$. Consequently, the smoothness of the kept part of the scaling function (the part inside the interval) becomes important, since heavy scaling will, when viewed on a fixed scale, potentially alter the smoothness drastically. This in turn results in arbitrarily large transform coefficients.

This is easily demonstrated with an example. We are satisfied at this point by considering the left edge only, and we choose the interval to be $[0; \infty)$. Different translations

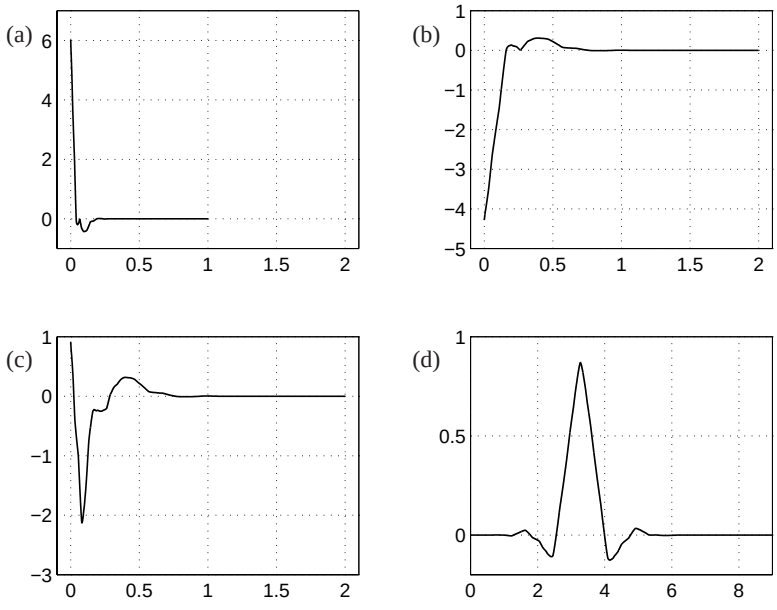


Figure 10.2: The result of restricting a scaling function to an interval followed by orthonormalization. The plots are described in the text.

of a scaling function will then give either exterior, interior, or edge scaling function. The latter are those translations which place the scaling function such that it's support is a subset of both $(-\infty; 0)$ and $(0; \infty)$. For instance, the Coiflet 12 scaling function, shown in Fig. 10.2(d), gives exactly 8 edge scaling functions since the support has measure 9. The left-most of those, i.e. the translated version with support $[-8; 1]$, is shown in plot (a). It has been restricted to the interval and normalized. The left-most but one is shown in plot (b). This, too, is restricted to the interval and normalized. The latter function is not orthogonal to the former. This is easily handled by the Gram-Schmidt procedure. The result (where (b) is orthonormalized with respect to (a) to preserve the staggered support) is shown in (c). Several more edge scaling functions have to be constructed before the transform is ready. But it is obvious just by looking at these plots that using this procedure for constructing edge scaling functions will result in a transform which is numerically unstable at the ends of the function.

In general, this method of construction produces edge scaling functions which have much faster high amplitude oscillations than the scaling function itself. The same oscillations are of course present in the tail of the scaling function, but with a much smaller amplitude. With longer scaling functions many edge scaling functions must be constructed, and the oscillatory behaviour will typically spread due to the orthonormalization.

Note that all scaling functions have connected support (see Lemarié-Rieusset and Malgouyres [52]), i.e. it cannot be vanishing (except in single points) inside the outer bounds of the support. There is therefore no reason to search for a scaling function with 'mostly vanishing' tail.

Moreover, the frequency interpretation of the wavelet basis also suffers under this construction. On the real line we can think of wavelets at a certain scale as a basis for representing a frequency band of approximately one octave. The highly oscillatory behaviour seen in Fig. 10.2 clearly cover many octaves. This concern is also expressed in Cohen et al. [22] at the end of Section 3 where the Meyer construction is discussed. This includes figures showing the full set of edge scaling functions for the Daubechies 4 and 8 scaling functions, which exhibits the same oscillatory behaviour as the Coiflet 12 edge scaling functions presented here.

10.2.3 Conclusion

The discussions throughout this section has shown that it is not a trivial task to base a construction of edge wavelets and edge scaling functions on polynomials. The reasons for investigating this subject nonetheless are 1) a lack of really suitable alternatives combined with 2) the nice properties of the low and high pass filters demonstrated in the beginning of this chapter.

The lack of alternative has not been explicitly stated, but the discussions in Chapter 9 on various methods for transforming finite signals hinted this, since this chapter presents the more well-known theoretical approaches to the problem of finite signals (at least to the best of the author's knowledge), and none of these includes the moment preserving

property of wavelets.

The nice polynomial-related properties of the wavelet filters was discussed in Section 10.1, and demonstrated explicitly in (10.3) and (10.4).

The task at hand is therefore to combine the moment preserving property of wavelets on the real line (or at least interior wavelets) with the idea of a transform that operates on a bounded interval of the real line. The difficulty in this task is basically how to construct the edge scaling functions such that they have not only the moment preserving property, but also the other nice properties that we expect of wavelets; orthonormality, staggered support, numerical stability, and agreement with the MRA. The latter property is important since this is the key to the filter taps. The following section describes in detail how to construct the edge functions to achieve all the desired properties simultaneously.

10.3 Construction of Moment Preserving Edge Filters

The process of constructing the edge filters to preserve both orthogonality and moments is not trivial, and requires a substantial number of computations. The process in divided into the following steps:

1. First we showed that polynomials of sufficiently low degree can be written as a linear combination of one layer of scaling functions (for fixed j), and that the convergence is nice on compact subsets of $\mathbb{R}$.
2. The next step is pursuing the idea mentioned in Section 10.1.2, that is constructing the edge functions one at a time, where each added function preserves one extra moment.
3. Changing the support of these new functions such that it becomes staggered leads to the definition of left edge scaling functions in Definition 10.2.
4. Theorem 10.3 shows that the left edge functions together with the interior scaling functions generate the desired polynomials.
5. Subsequently, Theorem 10.4 shows how to orthogonalize the left edge functions to make an orthonormal basis for $L^2([0;\infty))$, that the construction stays within the framework of MRA, and finally how to make the corresponding low pass filter taps.
6. The left edge wavelets are defined in a MRA sense in Lemma 10.5, and orthonormalized in Lemma 10.6, which also provides the high pass filter taps.
7. A necessary extra step in the transformation is introduced in Lemma 10.7, 10.8, and 10.9.

10.3.1 Constructing the New Edge Scaling Functions

The basic concept in the following approach is to construct the necessary scaling functions almost from scratch. Instead of modifying the edge functions that emerged naturally from looking at a compact subset I, we construct the edge functions specifically to be able to reproduce polynomials. All the other properties will be incorporated in the construction subsequently.

The edge functions are constructed one at a time, starting with the shortest (the left-most) scaling function. We want this function together with the interior scaling functions to generate the zeroth order polynomials, that is the constant functions, on I. To make things a little easier we first consider the interval $[0;\infty)$, that is the left edge only. We will later return to the right edge.

Now, to achieve the above mention property we define on $[0;\infty)$ the function

$$\tilde{\varphi}_0(t) \equiv 1 - \sum_{n=N-1}^{\infty} \phi(t-n) = \sum_{n=-\infty}^{N-2} \phi(t-n) = \sum_{n=-N+1}^{N-2} \phi(t-n). \tag{10.11}$$

Then $\phi_{0,m}$, $m \geq N-1$ together with $\tilde{\varphi}_0$ generates all constant functions on $[0;\infty)$. Note that $\tilde{\varphi}_0$ has compact support. It remains to ensure that this approach stays within the framework of the MRA. The two-scale equation for the translated ϕ is given by

$$\phi(t-n) = \sqrt{2} \sum_{m=2n-N+1}^{N+2n} h_{m-2n}\phi(2t-m). \tag{10.12}$$

Combining (10.11) and (10.12) yields

$$\begin{aligned}
\tilde{\varphi}_0(2^{j-1}t) &= 1 - \sum_{n=N-1}^{\infty} \phi(2^{j-1}t-n) && (10.13)\\
&= 1 - \sum_{n=N-1}^{\infty} \sqrt{2} \sum_{m=2n-N+1}^{N+2n} h_{m-2n}\phi(2^j t-m)\\
&= \tilde{\varphi}_0(2^j t) + \sum_{m=N-1}^{\infty} \phi(2^j t-m) - \sum_{m=N-1}^{\infty} \sqrt{2} \sum_{n=N-1}^{\infty} h_{m-2n}\phi(2^j t-m)\\
&= \tilde{\varphi}_0(2^j t) + \sum_{m=N-1}^{\infty} \Big[1 - \sqrt{2} \sum_{n=N-1}^{\infty} h_{m-2n}\Big]\phi(2^j t-m). && (10.14)
\end{aligned}$$

Hence

$$\operatorname{span}\{\tilde{\varphi}_{-j+1,0}, \phi_{-j+1,n}\}_{n\geq N-1} \subset \operatorname{span}\{\tilde{\varphi}_{-j,0}, \phi_{-j,n}\}_{n\geq N-1}, \tag{10.15}$$

since both sums in (10.14) are finite. This gives a series of spaces equivalent to V_j of the MRA. The next step is adding the 1st order polynomials. Using the same approach as before gives (here C_1 is chosen as the right constant)

$$\begin{aligned}
\tilde{\varphi}_1(t) &\equiv t - \sum_{n=N-1}^{\infty} n\phi(t-n) - C_1\\
&= \sum_{n} n\phi(t-n) + C_1 - \sum_{n=N-1}^{\infty} n\phi(t-n) - C_1
\end{aligned}$$

$$= \sum_{n=-\infty}^{N-2} n\phi(t-n)$$
$$= \sum_{n=-N+1}^{N-2} n\phi(t-n)\ .$$

Then

$$\begin{aligned}
2\tilde{\varphi}_1(2^{j-1}t) &= 2^j t - 2\sum_{n=N-1}^{\infty} n\phi(2^{j-1}t-n) - 2C_1 \\
&= \tilde{\varphi}_1(2^j t) + \sum_{n=N-1}^{\infty} n\phi(2^j t-n) - 2\sum_{n=N-1}^{\infty} n\phi(2^{j-1}t-n) - C_1 \\
&= \tilde{\varphi}_1(2^j t) + \sum_{n=N-1}^{\infty} n\phi(2^j t-n) - C_1\Big(\tilde{\varphi}_0(2^j t) + \sum_{n=N-1}^{\infty}\phi(2^j t-n)\Big) \\
&\quad - 2\sum_{n=N-1}^{\infty} n\sqrt{2}\sum_{m=2n-N+1}^{N+2n} h_{m-2n}\phi(2^j t-m) \\
&= \tilde{\varphi}_1(2^j t) + \sum_{n=N-1}^{\infty} n\phi(2^j t-n) - C_1\Big(\tilde{\varphi}_0(2^j t) + \sum_{n=N-1}^{\infty}\phi(2^j t-n)\Big) \\
&\quad - \sum_{n=N-1}^{\infty}\phi(2^j t-n)2\sqrt{2}\sum_{m=N-1}^{\infty} m h_{n-2m} \qquad (10.16) \\
&= \tilde{\varphi}_1(2^j t) - C_1\tilde{\varphi}_0(2^j t) \\
&\quad + \sum_{n=N-1}^{\infty}\Big(n - C_1 - 2\sqrt{2}\sum_{m=N-1}^{\infty} m h_{n-2m}\Big)\phi(2^j t-n) \qquad (10.17)
\end{aligned}$$

Note that in (10.16) the letters m and n have been interchanged in the last two sums. Now (10.14) and (10.17) gives

$$\text{span}\{\tilde{\varphi}_{-j+1,0}, \tilde{\varphi}_{-j+1,1}, \phi_{-j+1,n}\}_{n\geq N-1} \subset \text{span}\{\tilde{\varphi}_{-j,0}, \tilde{\varphi}_{-j,1}, \phi_{-j,n}\}_{n\geq N-1},$$

Again this implies the existence of a series of spaces equivalent to the V_j of the MRA. Finally, just to establish the pattern;

$$\tilde{\varphi}_2(t) = t^2 - \sum_{n=N-1}^{\infty} n^2\phi(t-n) - 2C_1 t + C_2 = \sum_{n=-N+1}^{N-2} n^2\phi(t-n).$$

It now seems obvious how to define the set of edge scaling functions (for the left edge). As is clear from the above, it is necessary with one edge function 'per degree'. With an

N vanishing moment scaling function it is possible to generate (in the pointwise sense) polynomials of degree $N-1$ on $\mathbb{R}$. In order to generate the same polynomials on $[0;\infty)$ a total of N edge function, at each edge, is needed. For j sufficiently large there is exactly $2^j - 2N + 2$ interior scaling function. Since the desired total number of scaling function is 2^j on level j, this leaves room for $N-1$ edge functions; one degree seems to be lost. To reclaim this degree the outermost interior scaling function is included in the definition of the edge scaling function. Thus, by defining

$$\tilde{\varphi}_k(t) = \sum_{n=-N+1}^{N-1} n^k \phi(t-n), \quad k = 0, \ldots, N-1\,, \tag{10.18}$$

we have a set of scaling functions which generates polynomials of degree $N-1$ on $[0;\infty)$.

10.3.2 The Desired Additional Properties

Unfortunately, the function defined in (10.18) all have the same support, which for a practical implementation gives a larger number of edge filter coefficients than the more direct Gram-Schmidt approach described in the previous chapter, and, more significantly, the resulting edge transform coefficients are subject to different interpretation that the interior transform coefficients. However, since

$$\begin{aligned}
\tilde{\varphi}_k(t) &= (-1)^k \sum_{n=-N+1}^{N-1} n^k \phi(t+n) \\
&= (-1)^k \sum_{n=0}^{2N-2} (n-N+1)^k \phi(t+n-N+1) \\
&= (-1)^k \sum_{n=0}^{2N-2} \sum_{m=0}^{k} \binom{k}{m} n^m (1-N)^{k-m} \phi(t+n-N+1) \\
&= (-1)^k \sum_{m=0}^{k} \binom{k}{m} (1-N)^{k-m} \sum_{n=0}^{2N-2} \sum_{u=0}^{m} \binom{n}{u} \tilde{\gamma}_{m,u} \phi(t+n-N+1) \\
&= \text{linear combination of } \sum_{n=0}^{2N-2} \binom{n}{u} \phi(t+n-N+1),
\end{aligned} \tag{10.19}$$

(where the second last equation follows from Lemma B.1, p. 311) and $\binom{n}{u} = 0$ for $n < u$, the property is restore by the following definition.

Definition 10.2 (Left Edge Scaling Functions)
For given N define the N edge scaling functions φ_k, $k = 0, \ldots, N-1$, on $[0; \infty)$ by

$$\varphi_k(t) = \sum_{n=k}^{2N-2} \binom{n}{k} \phi(t + n - N + 1). \tag{10.20}$$

The properties of these edge function can be summarize in the following theorem.

Theorem 10.3 (Generation of Polynomials by the New Edge Scaling Functions)
The N functions φ_k, $k = 0, \ldots, N-1$ are linearly independent, and orthogonal to $\phi_{0,m}$, $m \geq N$. Together with $\phi_{0,m}$, $m \geq N$ they generate all the polynomials up to degree $N-1$ on $[0; \infty)$. Finally,

$$\varphi_k(t) = \sum_{m=0}^{k} \alpha_{k,m} \varphi_k(2t) + \sum_{n=N}^{3N-2-2k} \beta_{k,n} \phi(2t - n), \tag{10.21}$$

where

$$\alpha_{k,n} = \frac{1}{\sqrt{2}} \sum_{q=n}^{k} 2^{-q} \gamma_{k,q} \sum_{r=n}^{q} \tilde{\gamma}_{r,n} \binom{q}{r} \sum_{m=0}^{q-r} \binom{q-r}{m} (N-1)^m \sum_{u=1-N}^{N} h_u u^{q-r-m} \tag{10.22}$$

$$\beta_{k,n} = \sqrt{2} \sum_{m=0}^{k} \gamma_{k,m} \sum_{s=1-N}^{N-1} (s + N - 1)^m h_{2s+n} \tag{10.23}$$

Proof
The linearly independency is immediate from the staggered support, which is $\operatorname{supp} \varphi_k = [0; 2N - 1 - k]$. The orthogonality with respect to the $\phi_{0,m}$, $m \geq N$ is also evident.

Proving that the edge functions generate the desired polynomials begins by returning to a slightly modified version of the original proposal (10.18) for edge functions. Define, again for $t \in [0; \infty)$,

$$\tilde{\varphi}_k(t) = \sum_{n=0}^{2N-2} n^k \phi(t + n - N + 1), \quad k = 0, \ldots, N-1. \tag{10.24}$$

Since these redefined $\tilde{\varphi}_k$ are linearly independent for $N \geq 2$ (due to the coefficients n^k), and since they, according to (10.19), span the same space as φ_k, it follows that the φ_k and $\phi_{0,m}$, $m \geq N$ generate all polynomials up to degree $N-$ if and only if the same holds for $\tilde{\varphi}_k$ and $\phi_{0,m}$, $m \geq N$. But applying the same trick as in (10.9) shows that the set of polynomials $p_k(t)$, $k = 0, \ldots, N-1$, with

$$p_k(t) \equiv \sum_n n^k \phi(t - n - N + 1)$$

$$= \sum_n [t - N + 1 - (t-n)]^k \phi(t-n)$$
$$= \sum_{m=0}^{k} \binom{k}{m} (-1)^m (t - N + 1)^{k-m} C_m,$$

generates all polynomials up to degree $N-1$ (since the leading terms of $p_k(t)$ is exactly t^k), and from (10.24) it is seen that for $t \in [0;\infty)$

$$p_k(t) = (-1)^k \tilde{\varphi}_k(t) + \sum_{n=1}^{\infty} n^k \phi(t - n - N + 1).$$

This implies that φ_k together with $\phi_{0,m}$, $m \geq N$, generate all polynomials up to degree $N-1$.

It now remains to establish the recurrence (10.21). From (10.12) and (10.24) follow (by substitution $m = s + 2n - 2N + 2$) that

$$\begin{aligned}
\tilde{\varphi}_k(t) &= \sum_{n=0}^{2N-2} n^k \sqrt{2} \sum_{m=-N+1}^{N} h_m \phi(2t + 2n - 2N + 2 - m) \\
&= \sqrt{2} \sum_{s=-3N+1}^{3N-2} \phi(2t-s) \sum_{n=0}^{2N-2} h_{2n-2N+2+s} n^k \\
&= \sqrt{2} \sum_{s=-N+1}^{N-1} \phi(2t-s) \sum_{n=0}^{2N-2} h_{2n-2N+2+s} n^k \qquad (10.25) \\
&\quad + \sqrt{2} \sum_{s=N}^{3N-2} \phi(2t-s) \sum_{n=-N+1}^{N-1} (n+N-1)^k h_{2n+s}\ , \qquad (10.26)
\end{aligned}$$

using that $t \geq 0$ for the last equality, and that $h_n = 0$ for $n < -N+1$ and $n > N$. The last part (10.26) is now on the right form. To rewrite (10.25), first note that

$$0 = m_0^{(r)}(\pi) = \frac{1}{\sqrt{2}} \sum_n h_n (-in)^r e^{-in\pi} = \sum_n h_n n^r (-1)^n (-i)^r,$$

for $r = 0, \ldots, N-1$, and hence

$$\begin{aligned}
\sum_n h_{n-m} (-1)^n n^r &= (-1)^m \sum_s h_s (-1)^s (s+m)^r \\
&= (-1)^m \sum_s h_s (-1)^s \sum_{u=0}^{r} \binom{r}{u} m^{r-u} s^u = 0.
\end{aligned}$$

Thus

$$0 = \sum_n h_{n-s}(-1)^n n^r = \sum_n h_{2n-s}(2n)^r - \sum_n h_{2n+1-s}(2n+1)^r,$$

such that

$$\sum_n h_{2n-s}(2n)^r = \sum_n h_{2n+1-s}(2n+1)^r = \frac{1}{2}\sum_n h_{n-s} n^r$$
$$= \frac{1}{2}\sum_n h_n \sum_{m=0}^{r}\binom{r}{m} s^m n^{r-m} = \frac{1}{\sqrt{2}}\sum_{m=0}^{r}\binom{r}{m} s^m M_{r-m}.$$

Then the last sum in (10.25) can be substituted by

$$\sum_n h_{2n-2N+2+s} n^k = \sum_n 2^{-k} h_{2n-N+1-(N-1-s)}(2n)^k$$
$$= \sum_n 2^{-k} h_{2n-N+1-(N-1-s)}\big[2n + (N-1-s) - (N-1-s)\big]^k$$
$$= \sum_{r=0}^{k}\binom{k}{r}(N-1-s)^r 2^{-k}\sum_n h_{2n-N+1-(N-1-s)}\big(2n-(N-1-s)\big)^{k-r}$$
$$= \frac{1}{\sqrt{2}}\sum_{r=0}^{k}\binom{k}{r}(N-1-s)^r 2^{-k}\sum_{m=0}^{k-r}\binom{k-r}{m}(N-1)^m M_{k-r-m},$$

such that

$$(10.25) = \sum_{s=-N+1}^{N-1}\phi(2t+s)\sum_{r=0}^{k}\binom{k}{r}(s+N-1)^r 2^{-k}\sum_{m=0}^{k-r}\binom{k-r}{m}(N-1)^m M_{k-r-m}$$
$$= \sum_{r=0}^{k}\tilde{\varphi}_r(2t)\binom{k}{r}2^{-k}\sum_{m=0}^{k-r}\binom{k-r}{m}(N-1)^m M_{k-r-m}.$$

By applying lemma B.1, see p. 311 to (10.24) and (10.20), respectively, yields

$$\tilde{\varphi}_k(t) = \sum_{m=0}^{k}\tilde{\gamma}_{k,m}\varphi_m(t) \qquad \text{and} \qquad \varphi_k(t) = \sum_{m=0}^{k}\gamma_{k,m}\tilde{\varphi}_m(t).$$

Using this on (10.25)/(10.26) with the above rewriting yields

$$\varphi_k(t) = \sum_{q=0}^{k}\gamma_{k,q}\sum_{r=0}^{q}\sum_{u=0}^{r}\tilde{\gamma}_{r,u}\varphi_u(2t)\binom{q}{r}2^{-q}\sum_{m=0}^{q-r}\binom{q-r}{m}(N-1)^m M_{q-r-m}$$

$$+\sqrt{2}\sum_{q=0}^{k}\gamma_{k,q}\sum_{s=N}^{3N-2}\phi(2t-s)\sum_{n=-N+1}^{N-1}(n+N-1)^q h_{2n+s}. \quad (10.27)$$

Define now

$$d_{q,r}=2^{-q}\binom{q}{r}\sum_{m=0}^{r}\binom{r}{m}(N-1)^m M_{q-r-m}.$$

Then

$$\begin{aligned}\varphi_k(t)&=\sum_{q=0}^{k}\gamma_{k,q}\sum_{r=0}^{q}d_{q,r}\sum_{u=0}^{r}\tilde{\gamma}_{r,u}\varphi_u(2t)+\sum_{s=N}^{3N-2}\beta_{k,s}\phi(2t-s)\\&=\sum_{q=0}^{k}\gamma_{k,q}\sum_{u=0}^{q}\varphi_u(2t)\sum_{r=u}^{q}d_{q,r}\tilde{\gamma}_{r,u}+\sum_{s=N}^{3N-2}\beta_{k,s}\phi(2t-s)\\&=\sum_{u=0}^{k}\varphi_u(2t)\sum_{q=u}^{k}\gamma_{k,q}\sum_{r=u}^{q}d_{q,r}\tilde{\gamma}_{r,u}+\sum_{s=N}^{3N-2}\beta_{k,s}\phi(2t-s)\\&=\sum_{u=0}^{k}\alpha_{k,u}\varphi_u(2t)+\sum_{s=N}^{3N-2}\beta_{k,s}\phi(2t-s)\end{aligned}$$

□

This provides a set of edge scaling functions which together with the interior scaling functions generate the desired polynomials. But they do not provide an orthonormal basis, since the φ's are not orthogonal. This, however, is achievable through a linear mapping of the φ's.

Theorem 10.4 (Orthonormal Left Edge Scaling Functions and Filter Taps)
There exists an $N\times N$ invertible matrix $\tilde{\mathbf{E}}=\mathbf{E}^{-1}$ such that $\varphi_k^{\text{left}}(t)$, $k=0,\ldots,N-1$, defined by

$$\begin{bmatrix}\varphi_{N-1}^{\text{left}}\\ \vdots\\ \varphi_1^{\text{left}}\\ \varphi_0^{\text{left}}\end{bmatrix}\equiv\tilde{\mathbf{E}}\begin{bmatrix}\varphi_0\\ \vdots\\ \varphi_{N-2}\\ \varphi_{N-1}\end{bmatrix}, \quad (10.28)$$

1. *have support $[0;N+k]$,*
2. *together with $\phi_{0,m}$, $m\geq N$ generate all the polynomials up to degree $N-1$ on $[0;\infty)$,*
3. *have the property that*

$$\{\varphi_{-j,k}^{\text{left}}\}_{k=0,\ldots,N-1}\cup\{\phi_{-j,m}\}_{m\geq N} \quad (10.29)$$

is an orthonormal set with the property that

$$V_j^{\text{left}} \equiv \overline{\text{span}\big[\{\varphi_{-j,k}^{\text{left}}\}_{k=0,\ldots,N-1} \cup \{\phi_{-j,m}\}_{m\geq N}\big]} \tag{10.30}$$

satisfy

$$\cdots \subset V_2^{\text{left}} \subset V_1^{\text{left}} \subset V_0^{\text{left}} \subset V_{-1}^{\text{left}} \subset V_{-2}^{\text{left}} \subset \cdots \tag{10.31}$$

and $\cap_j V_j^{\text{left}} = \{0\}$ *and* $\overline{\cup_j V_j^{\text{left}}} = L^2\big([0;\infty)\big)$,

4. *satisfy the recurrence*

$$\varphi_{j,m}^{\text{left}} = \sum_{s=0}^{N-1} h_{m,s}^{left} \varphi_{j-1,s}^{\text{left}} + \sum_{s=N}^{N+2m} h_{m,s}^{left} \phi_{j-1,s}, \tag{10.32}$$

where

$$h_{m,s}^{left} = \frac{1}{\sqrt{2}} \sum_{n=0}^{N-1} e_{n,N-1-s} \sum_{k=n}^{N-1} \tilde{e}_{N-1-m,k} \alpha_{k,n} \qquad \text{for } s < N, \tag{10.33}$$

$$h_{m,s}^{left} = \frac{1}{\sqrt{2}} \sum_{k=N-1-m}^{N-1} \tilde{e}_{N-1-m,k} \beta_{k,s} \qquad \text{for } s \geq N. \tag{10.34}$$

Proof

To orthonormalize the φ's the overlap matrix $\mathbf{E} = [\eta_{k,n}] \equiv [\langle \varphi_k, \varphi_n \rangle]$ is needed first of all. By the recurrence (10.21)

$$\begin{aligned} 2\eta_{k,s} &= \sum_{m=0}^{k-1} \sum_{n=0}^{s} \alpha_{k,m}\alpha_{s,n}\eta_{m,n} + \sum_{n=0}^{s-1} \alpha_{k,k}\alpha_{s,n}\eta_{k,n} + \alpha_{k,k}\alpha_{s,s}\eta_{k,s} + \sum_{m=N}^{3N-2-2k} \beta_{k,m}\beta_{s,m} \\ &= \frac{2}{2-2^{-k-s}} \Big(\sum_{m=0}^{k-1} \sum_{n=0}^{s} \alpha_{k,m}\alpha_{s,n}\eta_{m,n} + \sum_{n=0}^{s-1} 2^{-k}\alpha_{s,n}\eta_{k,n} + \sum_{m=N}^{3N-2-2k} \beta_{k,m}\beta_{s,m} \Big). \end{aligned}$$

Note that according to (10.22) $\alpha_{k,k} = 2^{-k}$. Now it is possible to determine $\eta_{k,s}$ for $s = 0, \ldots, k$ (in that order) when $\eta_{m,n}$ is known for $m, n = 0, \ldots, k-1$ with $n \leq m$. To preserve the staggered support orthogonalization begins with the last edge scaling function φ_{N-1}. The Gram-Schmidt procedure is in that case

$$\varphi_k^{\times}(t) = \varphi_k(t) - \sum_{n=k+1}^{N-1} \frac{\langle \varphi_k^{\times}, \varphi_n^{\times} \rangle}{\langle \varphi_n^{\times}, \varphi_n^{\times} \rangle} \varphi_n^{\times}(t). \tag{10.35}$$

In each step this procedure relies on previously orthogonalized functions (which is denoted $\varphi^{\times}$), and therefore the 'orthogonalized' η's are also need. Define therefore

$$\tilde{\eta}_{k,n} = \begin{cases} \langle \varphi_k, \varphi_n^{\times} \rangle & \text{for } n > k \\ \langle \varphi_k^{\times}, \varphi_k^{\times} \rangle & \text{for } n = k. \end{cases}$$

The link to the η's is the recursive equation

$$\tilde{\eta}_{k,n} = \langle \varphi_k, \varphi_n \rangle - \sum_{s=n+1}^{N-1} \frac{\langle \varphi_n, \varphi_s^{\times} \rangle}{\langle \varphi_s^{\times}, \varphi_s^{\times} \rangle} \langle \varphi_k, \varphi_s^{\times} \rangle = \eta_{k,n} - \sum_{s=n+1}^{N-1} \frac{\tilde{\eta}_{n,s}\tilde{\eta}_{k,s}}{\tilde{\eta}_{s,s}}, \qquad k \le n. \quad (10.36)$$

Given all $\tilde{\eta}_{k,n}$ for $n > m$, (10.36) determines $\tilde{\eta}_{k,m}$ for $k = 0, \ldots, m$ (in no particular order). The orthogonalization can now be carried out in the following manner. Define the $N \times N$ matrix

$$\tilde{\mathbf{E}}_k = \mathbf{I}_{N\times N} + \begin{bmatrix} \mathbf{0}_{k\times N} \\ -\tilde{\mathbf{e}}_k \\ \mathbf{0}_{N-k-1\times N} \end{bmatrix}, \quad (10.37)$$

where

$$\tilde{\mathbf{e}}_k = \begin{bmatrix} 0 & \cdots & 0 & \dfrac{\tilde{\eta}_{k,k+1}}{\tilde{\eta}_{k+1,k+1}} & \cdots & \dfrac{\tilde{\eta}_{k,N-1}}{\tilde{\eta}_{N-1,N-1}} \end{bmatrix}, \quad k = 0, \ldots, N-1, \quad (10.38)$$

that is starting with $k+1$ zeros. Define also $\boldsymbol{\varphi} = \begin{bmatrix} \varphi_0 & \ldots & \varphi_{N-1} \end{bmatrix}^{\top}$. From (10.35) now follows that

$$\begin{bmatrix} \varphi_0 \\ \vdots \\ \varphi_{N-2} \\ \varphi_{N-1}^{\times} \end{bmatrix} = \tilde{\mathbf{E}}_{N-1}\boldsymbol{\varphi} \ ,$$

and using (10.35) once more yields

$$\begin{bmatrix} \varphi_0 \\ \vdots \\ \varphi_{N-3} \\ \varphi_{N-2}^{\times} \\ \varphi_{N-1}^{\times} \end{bmatrix} = \tilde{\mathbf{E}}_{N-2}\tilde{\mathbf{E}}_{N-1}\boldsymbol{\varphi} \ .$$

Since the $\varphi^{\times}$'s are orthogonal only the normalization remains. Hence the complete orthonormalization is given by

$$\begin{bmatrix} \varphi_{N-1}^{\text{left}} \\ \vdots \\ \varphi_1^{\text{left}} \\ \varphi_0^{\text{left}} \end{bmatrix} = \begin{bmatrix} \tilde{\eta}_{0,0}^{-1/2} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \tilde{\eta}_{N-1,N-1}^{-1/2} \end{bmatrix} \prod_{m=0}^{N-1} \tilde{\mathbf{E}}_m \begin{bmatrix} \varphi_0 \\ \vdots \\ \varphi_{N-2} \\ \varphi_{N-1} \end{bmatrix} = \tilde{\mathbf{E}}\boldsymbol{\varphi} \ . \quad (10.39)$$

The φ^{left}'s are the orthonormalize edge scaling functions. Note that these are indexed reversely. This is to retain the 'shift of support' structure, which exists for the interior scaling functions, i.e. the support of φ_k^{left} increases to the right with increasing k. In

order to use these scaling functions the corresponding filters are needed. This is done via a recurrence equivalent to (10.21), but with φ^{left} instead of φ. Expanding the m'th orthonormalized scaling function yields

$$\begin{aligned}\varphi^{\text{left}}_{j,N-1-m} &= \sum_{k=m}^{N-1} \tilde{e}_{m,k}\varphi_{-j,k} \\ &= \frac{1}{\sqrt{2}} \sum_{n=0}^{N-1}\sum_{k=n}^{N-1} \tilde{e}_{m,k}\alpha_{k,n}\varphi_{j-1,n} + \sum_{k=m}^{N-1} \tilde{e}_{m,k} \sum_{s=N}^{3N-2-2k} \beta_{k,s}\phi_{j-1,s} \ .\end{aligned}$$

Then

$$\begin{aligned}\varphi^{\text{left}}_{j,m} &= \frac{1}{\sqrt{2}} \sum_{n=0}^{N-1} \varphi_{j-1,n} \sum_{k=n}^{N-1} \tilde{e}_{N-1-m,k}\alpha_{k,n} \\ &\qquad + \frac{1}{\sqrt{2}} \sum_{k=N-1-m}^{N-1} \tilde{e}_{N-1-m,k} \sum_{s=N}^{3N-2-2k} \beta_{k,s}\phi_{j-1,s} \\ &= \frac{1}{\sqrt{2}} \sum_{n=0}^{N-1}\sum_{u=0}^{N-1} e_{n,N-1-u}\varphi^{\text{left}}_{j-1,u} \sum_{k=n}^{N-1} \tilde{e}_{N-1-m,k}\alpha_{k,n} \\ &\qquad + \frac{1}{\sqrt{2}} \sum_{s=N}^{N+2m} \sum_{k=N-1-m}^{N-1} \tilde{e}_{N-1-m,k}\beta_{k,s}\phi_{j-1,s} \\ &= \sum_{u=0}^{N-1} h^{\text{left}}_{m,u}\varphi^{\text{left}}_{j-1,u} + \sum_{s=N}^{N+2m} h^{\text{left}}_{m,s}\phi_{j-1,s} \ ,\end{aligned}$$

where $h^{\text{left}}_{m,u}$ are given by (10.33) and $h^{\text{left}}_{m,s}$ are given by (10.34). □

This concludes the construction of left edge scaling functions, and the attention now turns to constructing a corresponding set of edge wavelets. As with the MRA on the real line the scaling functions generates a set of spaces V^{left}_j, and it is therefore natural to define the edge scaling functions based on the 'difference space' $V^{\text{left}}_{j-1} \ominus V^{\text{left}}_j$, or, more elaborately, $W^{\text{left}}_j \equiv V^{\text{left}}_{j-1} \cap (V^{\text{left}}_j)^{\perp}$. The $\psi_{j,m}$, $m \geq N$, all belong to W^{left}_j, so N extra functions in W^{left}_j orthogonal to these $\psi_{j,m}$ are needed. Since the focus is on the difference between two successive V^{left}, an obvious definition of the N extra functions would be:

Lemma 10.5 (MRA Construction of Left Edge Wavelets)
Define the function ψ_k, $k = 0, \ldots, N-1$, by

$$\psi_k \equiv \varphi^{\text{left}}_{-1,k} - \sum_{m=0}^{N-1} \left\langle \varphi^{\text{left}}_{-1,k}, \varphi^{\text{left}}_{0,m} \right\rangle \varphi^{\text{left}}_{0,m} \ . \tag{10.40}$$

Then the ψ_k are N linearly independent functions in W_0^{left}, and orthogonal to the $\psi_{0,m}$, $m \geq N$.

Proof
Since the $\phi_{0,m}$, $\psi_{0,m}$, $m \geq N$ are linear combinations of the $\phi_{-1,m}$, $m \geq N+1$, which in turn are orthogonal to $\varphi_{-1,k}^{\text{left}}$, $k = 0, \ldots, N-1$, the ψ_k are orthogonal projections of $\varphi_{-1,k}^{\text{left}}$ onto W_0^{left}, and, being linear combinations of functions orthogonal to $\psi_{0,m}$, the ψ_k are obviously orthogonal to $\psi_{0,m}$, $m \geq N$.

To establish the linear independence, note that

$$\operatorname{supp} \varphi_{-1,k} = [0;\, N/2 + k/2] \qquad \text{and} \qquad \operatorname{supp} \varphi_{0,k} = [0;\, N + k],$$

which implies that $\varphi_{-1,k}$ and $\varphi_{0,k}$ are $2N$ linearly independent functions. Substituting ψ_k into

$$\sum_{s=0}^{N-1} a_s \psi_s + \sum_{n=0}^{N-1} \beta_n \varphi_{0,n}^{\text{left}} = 0 \tag{10.41}$$

gives

$$\sum_{s=0}^{N-1} a_s \varphi_{-1,s}^{\text{left}} + \sum_{n=0}^{N-1} \varphi_{0,n}^{\text{left}} \Big(\beta_n - \sum_{s=0}^{N-1} a_s \left\langle \varphi_{-1,s}^{\text{left}}, \varphi_{0,n}^{\text{left}} \right\rangle \Big) = 0$$

This holds if and only if all $a_s = 0$. Then by (10.41) all $\beta_n = 0$, showing that ψ_k and $\varphi_{0,k}^{\text{left}}$ are $2N$ independent functions. Hence ψ_k are N independent functions. □

This way of constructing the ψ_k does not give them staggered support. But since they are linearly independent this can be done through Gaussian elimination. And it turns out that the nice structure of the edge scaling function filter coefficients is preserved for the edge wavelet coefficients.

Lemma 10.6 (Orthonormal Left Edge Wavelets and Filter Taps)
There exists a linear map $\mathbf{L}$ *such that* ψ_k^{left}, $k = 0, \ldots, N-1$, *defined by*

$$\begin{bmatrix} \psi_0^{\text{left}} \\ \vdots \\ \psi_{N-2}^{\text{left}} \\ \psi_{N-1}^{\text{left}} \end{bmatrix} \equiv \mathbf{L} \begin{bmatrix} \psi_0 \\ \vdots \\ \psi_{N-2} \\ \psi_{N-1} \end{bmatrix}, \tag{10.42}$$

have support $[0;\, N+k]$, *and have the property that*

$$\{\psi_{-j,k}^{\text{left}}\}_{k=0,\ldots,N-1} \cup \{\psi_{-j,m}\}_{m \geq N} \tag{10.43}$$

is an orthonormal set. Moreover there exists constants $g_{k,s}^{\text{left}}$, $k = 0, \ldots, N-1$, $s = 0, \ldots, 3N-2$, *such that*

$$\psi_{-j,m}^{\text{left}} = \sum_{s=0}^{N-1} g_{m,s}^{\text{left}} \varphi_{-j-1,s}^{\text{left}} + \sum_{s=N}^{N+2m} g_{m,s}^{\text{left}} \phi_{-j-1,s}. \tag{10.44}$$

The proof is made short by postponing the actual calculations.

Proof

From (10.32) it follows that there exists constants $d_{k,m}$ such that

$$\psi_k = \sum_{u=0}^{N-1} d_{k,u}\varphi_{-1,u}^{\text{left}} + \sum_{u=N}^{3N-2} d_{k,u}\phi_{-1,u}, \tag{10.45}$$

Since the ψ_k are all orthogonal to $\phi_{0,N+n}$ for $n = 0, \ldots, N-2$,

$$\begin{aligned} 0 &= \langle \psi_k, \phi_{0,N+n} \rangle \\ &= \sum_{u=N}^{3N-2} d_{k,u} \sum_{m=-N+1}^{N} h_m \langle \phi_{-1,u}, \phi_{-1,m+2N+2n} \rangle \\ &= \sum_{u=N+1+2n}^{3N-2} d_{k,u} h_{u-2N-2n}. \end{aligned} \tag{10.46}$$

Now, if $d_{k,3N-2} = 0$ for all k, let $\tilde{\psi}_{N-1} = \psi_{N-1}$. But if $d_{k,3N-2} \neq 0$ for some k, reorder the ψ_k so that $d_{N-1,3N-2} \neq 0$, and let $\tilde{\psi}_{N-1} = \psi_{N-1}$, and for $k \leq N-2$, let

$$\psi_k^{(1)} = \psi_k - \frac{d_{k,3N-2}}{d_{N-1,3N-2}}\psi_{N-1}.$$

Since from (10.46) with $n = N-2$

$$d_{k,3N-3}h_{-N+1} + d_{k,3N-2}h_{-N+2} = 0,$$

it is clear that when coefficient to $\phi_{-1,3N-2}$ is zero, so is the coefficient to $\phi_{-1,3N-3}$. Hence $\psi_k^{(1)}$, $k = 0, \ldots, N-2$ satisfy a recursion relation similar to (10.45) with the upper limit $3N-4$. Consequently the support of $\psi_k^{(1)}$ is in $[0; 2N-2]$. Repeating this $N-1$ times yields the staggered support.

By orthonormalizing the $\tilde{\psi}_k$ via the Gram-Schmidt procedure starting with $k = 0$ (the smallest support), the result is an orthonormal set ψ_k^{left}, $k = 0, \ldots, N-1$ with staggered support. For any $j \in \mathbb{Z}$ define $\psi_{-j,k}^{\text{left}}(t) = 2^{j/2}\psi_k^{\text{left}}(2^j t)$. Together with $\psi_{-j,m}$, $m \geq N$, they (by construction) provide an orthonormal basis for W_j^{left}. □

The actual calculations are quite easy, because the ψ_k^{left} does not, as opposed to the edge scaling functions, dependent on themselves. First note that

$$\begin{aligned} \psi_k &= \varphi_{-1,k}^{\text{left}} - \sum_{m=0}^{N-1} h_{m,k}^{\text{left}} \Big(\sum_{u=0}^{N-1} h_{m,u}^{\text{left}} \varphi_{-1,u}^{\text{left}} + \sum_{u=N}^{N+2m} h_{m,u}^{\text{left}} \phi_{-1,u} \Big) \\ &= \varphi_{-1,k}^{\text{left}} - \sum_{u=0}^{N-1} \varphi_{-1,u}^{\text{left}} \sum_{m=0}^{N-1} h_{m,k}^{\text{left}} h_{m,u}^{\text{left}} - \sum_{u=N}^{3N-2} \phi_{-1,u} \sum_{m=0}^{N-1} h_{m,k}^{\text{left}} h_{m,u}^{\text{left}} \end{aligned}$$

$$= \sum_{u=0}^{N-1} d_{k,u}\varphi^{\text{left}}_{-1,u} + \sum_{u=N}^{3N-2} d_{k,u}\phi_{-1,u},$$

where $\mathbf{D} \equiv \mathbf{I} - (\mathbf{H}^{\text{left}}_N)^\top \mathbf{H}^{\text{left}}_N$ and

$$\mathbf{H}^{\text{left}}_N \equiv \begin{bmatrix} h^{\text{left}}_{0,0} & \cdots & h^{\text{left}}_{0,N} & & & & 0 \\ h^{\text{left}}_{1,0} & \cdots & h^{\text{left}}_{1,N} & h^{\text{left}}_{1,N+1} & h^{\text{left}}_{1,N+2} & & \\ \vdots & & \vdots & & & \ddots & \\ h^{\text{left}}_{N-1,0} & \cdots & h^{\text{left}}_{N-1,N} & h^{\text{left}}_{N-1,N+1} & \cdots & h^{\text{left}}_{N-1,3N-3} & h^{\text{left}}_{N-1,3N-2} \end{bmatrix}. \tag{10.47}$$

For future reference, define $\mathbf{G}^{\text{left}}_N$ equivalently. Since $\mathbf{D}$ has full row rank it is possible through Gaussian elimination to produce a 'lower triangular' matrix. This procedure is standard, and will not be explicitly presented here. The following orthonormalization can now be carried out directly on the lower triangular matrix, starting from the top to preserved the staggered support.

This concludes the derivation of edge functions and edge filters.

10.4 Conditioning

Now that the edge filters have been constructed, the next step is to see them in action. Using, for instance, the Daubechies length 8 filter (see Table 10.1) to transform an arbitrarily chosen third degree polynomial we are in for a surprise. A third degree polynomial is shown in Fig. 10.3(a) and the transform into low and high pass part is shown in (b). The low pass part is obviously not a sampled third degree polynomials (which it ought to be, according to the Lemma 10.7), nor is the high pass part the zero sequence (although this is supposed to be one of the key features of the edge filters). Alas, there seems to be a malfunction somewhere in the construction laid out in the previous section.

Fortunately, this is not the case. But the theory needs to be extended to handle this problem. This section is dedicated to a thorough description of how to do this. The first step is to establish that what we expected on the interval, is indeed true on the real line. Namely that the wavelet transform of a polynomial (of sufficiently low degree) will produce a new polynomial of the same degree, and reversely that any polynomial of sufficiently low degree is the wavelet transform of some other polynomial of the same degree.

Then Lemma 10.8 shows that although this does not hold on the interval, the edge filter transform is indeed a mapping from a vector space $\mathbb{S}'$ onto itself. It is thus possible to construct a mapping $\mathbf{A}$ between that space and the space $\mathbb{S}$ of sampled polynomials. This is done in Lemma 10.9 and Lemma 10.10. The principle is depicted below. The mapping $\mathbf{B}$ is the desired one, but, as Fig. 10.3 demonstrated, it is unfortunately not the same as the edge filter DWT. Thus, the mapping $\mathbf{A}$ is needed to move the signal between

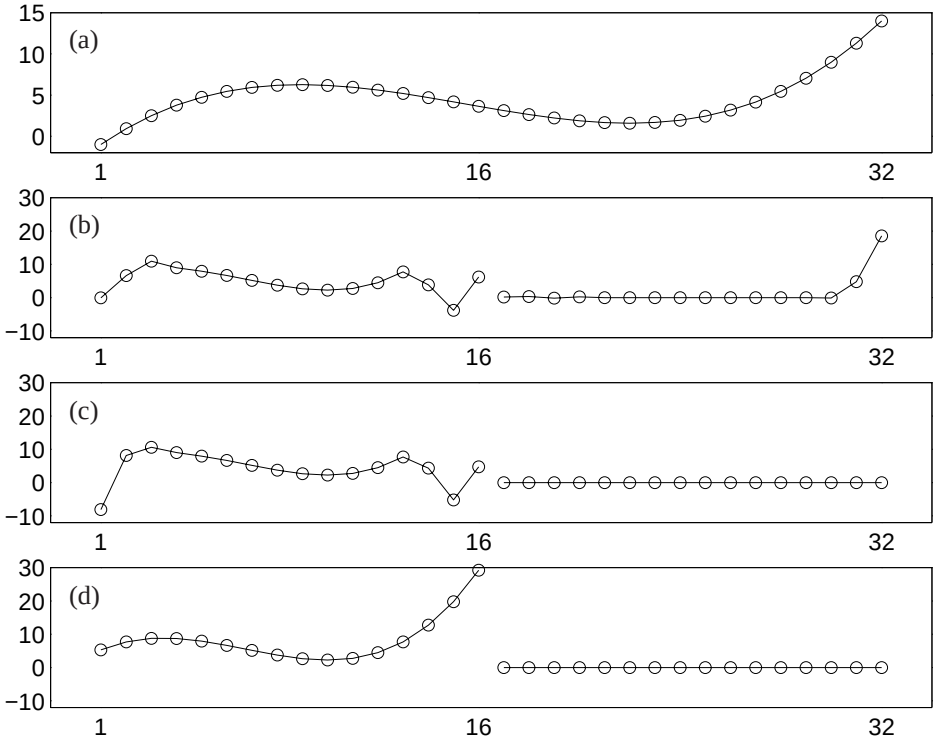
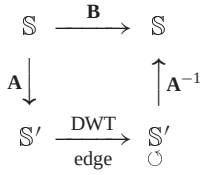


Figure 10.3: In (a) the polynomial $t^3 - 4t^2 + 2t + 6$ is sampled in 32 points in the interval $[-1; 4]$. When applying the DWT using the low and high pass edge filters derived in the previous sections, the result (b) is not exactly as expected. Daubechies length 8 filters are used. When applying the preconditioning matrices the high pass part becomes vanishing, showed in (c), while the low pass part needs an additional postconditioning before it becomes a third degree polynomial sampled in 16 points, as is seen in (d).

the two spaces.



It is hinted in Fig. 10.3(b) that **A** should only affect the ends of the signal. As this is indeed the case, the mapping is divided into two parts, one for each end. This is made explicit in Lemma 10.9.

10.4.1 Identifying the Problem

First, this lemma establishes that the approximation by $\phi(t-n)$ on the real line is a mapping from the space low ordered polynomials onto the space of sampled equally low ordered polynomials.

Lemma 10.7 (Mapping of Polynomials on the Real Line)
Let ϕ be a scaling function with $m_0^{(s)}(\pi) = 0$, $s = 0, \ldots, N-1$. Let $k \leq N-1$ and $p_k(t) \in \mathcal{P}_k(\mathbb{R})$ have of degree k, and let $\mathbf{a} \in \mathbb{R}^{k+1}$, $a_k \neq 0$. Then there exists a $p'_k(t) \in \mathcal{P}_k(\mathbb{R})$ of degree exactly k and $\mathbf{b} \in \mathbb{R}^{k+1}$ such that

$$\int_{\mathbb{R}} p_k(t)\phi(t-x)dt = \sum_{s=0}^{k} b_s x^s, \tag{10.48}$$

and

$$\sum_{s=0}^{k} a_s x^s = \int_{\mathbb{R}} p'_k(t)\phi(t-x)dt\ , \tag{10.49}$$

both for all $x \in \mathbb{R}$.

Note that although the integration in the lemma and the proof is over $\mathbb{R}$ (implicitly), all the integrands are compactly supported functions. Note also that the equalities are, as demonstrated in Section 10.2, only valid pointwise.

Proof
The first equation (10.48) follows immediately from

$$\int t^k \phi(t-x)dt = \int (t+x)^k \phi(t)dt = \int \sum_{m=0}^{k} \binom{k}{m} t^m x^{k-m} \phi(t)dt$$
$$= \sum_{m=0}^{k} \binom{k}{m} x^{k-m} \int t^m \phi(t)dt\ .$$

For the second equation let $p'_k(t) = \sum_{m=0}^{k} b'_m t^m$. Then

$$\int p'_k(t)\phi(t-x)dt = \int \sum_{m=0}^{k} b'_m t^m \phi(t-x)dt = \sum_{m=0}^{k} b'_m \sum_{s=0}^{m} x^{m-s}\binom{m}{s}\int t^s\phi(t)dt$$
$$= \sum_{m=0}^{k} b'_m \sum_{s=0}^{m} x^{m-s} B_{m,s} = \sum_{m=0}^{k} x^m \sum_{s=m}^{k} b'_s B_{s,s-m} \ .$$

Define now

$$\mathbf{B} = \begin{bmatrix} B_{0,0} & B_{1,1} & \dots & B_{k,k} \\ & B_{1,0} & \dots & B_{k,k-1} \\ & & \ddots & \vdots \\ 0 & & & B_{k,0} \end{bmatrix}, \qquad B_{m,s} = \binom{m}{s}\int t^s\phi(t)dt$$

then $\mathbf{B}$ is invertible, since $B_{m,0} = 1$, and hence choosing $\mathbf{b}' = \mathbf{B}^{-1}\mathbf{a}$ gives the coefficients of the polynomial $p'_k(t)$. □

As a consequence of this lemma sequences originating from equidistantly sampled polynomials of sufficiently low degree gets mapped to zero under the high pass filtering, since for any such sequence c_n, $n \in \mathbb{Z}$, one can, according to (10.49), find a polynomial $p(t)$ such that $c_n = \int p(t)\phi(t-n)dt$. Thus (and this has actually already been demonstrated in (10.3))

$$\sum_n g_{n-2k}c_n = \int p(t)\sum_n g_{n-2k}\phi(t-n)dt = \int p(t)2^{-1/2}\psi(t/2-k)dt = 0 \ .$$

In particular, the sequence $\{\int \phi(t-n)\}_{n\in\mathbb{Z}}$, which is just all 1's, maps to the zero sequence under high pass filtering. This is still true on the interval [0; 1], where

$$\begin{bmatrix} \int \varphi^{\text{left}}_{-j,0} & \dots & \int \varphi^{\text{left}}_{-j,N-1} & \int \phi_{-j,N} & \dots & \int \phi_{-j,2^j-N-1} \\ & & & & \int \varphi^{\text{right}}_{-j,-N+1} & \dots & \int \varphi^{\text{right}}_{-j,0} \end{bmatrix} \quad (10.50)$$

maps to the zero sequences (see Lemma 10.8). However, the sequence (10.50) is no longer a sequence consisting of just 1's, since the edge functions do not have integral 1! For instance, for the Daubechies 8 filter the edge functions integrate like

n	$\int \varphi^{\text{left}}_n(t)dt$	n	$\int \varphi^{\text{right}}_n(t)dt$
0	1.443	0	1.000
1	1.373	−1	0.347
2	1.209	−2	-0.443
3	1.032	−3	0.434

However, the following lemma shows that although the transform with edge filters maps sequences on the form (10.50) to zeros (instead of mapping sampled polynomials to zero), it is at least well-behaved in the sense that it maps these sequences to the same type of sequences.

Lemma 10.8
Let ϕ be a scaling function with with $m_0^{(s)}(\pi) = 0$, $s = 0, \ldots, N-1$, and $p(t)$ a polynomial of degree at most $N-1$, and let φ_k^{left}, $k = 0, \ldots, N-1$, be edge scaling functions as defined in Theorem 10.4. Define

$$\mathbf{c} = \begin{bmatrix} \langle p, \varphi_{-j,0}^{\text{left}} \rangle & \ldots & \langle p, \varphi_{-j,N-1}^{\text{left}} \rangle & \langle p, \phi_{-j,N} \rangle & \ldots & \langle p, \phi_{-j,3N-2} \rangle \end{bmatrix}^\top \tag{10.51}$$

Then

$$\mathbf{H}_N^{\text{left}} \mathbf{c} = \begin{bmatrix} \langle p, \varphi_{-j+1,0}^{\text{left}} \rangle & \ldots & \langle p, \varphi_{-j+1,N-1}^{\text{left}} \rangle \end{bmatrix}^\top \tag{10.52}$$

and

$$\mathbf{G}_N^{\text{left}} \mathbf{c} = \mathbf{0} \, . \tag{10.53}$$

Proof
From (10.32) it follows that

$$\sum_{m=0}^{N-1} h_{k,m}^{\text{left}} \langle p, \varphi_{-j,m}^{\text{left}} \rangle + \sum_{m=N}^{N+2k} h_{k,m}^{\text{left}} \langle p, \phi_{-j,m} \rangle = \langle p, \varphi_{-j+1,k}^{\text{left}} \rangle \, ,$$

which proves (10.52). Likewise, from (10.44)

$$\sum_{m=0}^{N-1} g_{k,m}^{\text{left}} \langle p, \varphi_{-j,m}^{\text{left}} \rangle + \sum_{m=N}^{N+2k} g_{k,m}^{\text{left}} \langle p, \phi_{-j,m} \rangle = \langle p, \psi_{-j+1,k}^{\text{left}} \rangle \, ,$$

and (10.53) then follows from

$$\int t^k \psi_{-j,m}^{\text{left}}(t) dt = 0, \qquad k = 0, \ldots, N-1 \, .$$

□

10.4.2 Constructing the **A** Mapping

So, one way to handle this problem is to convert polynomials sequences $\sum_{s=0}^{k} b_s n^s$ into sequences on the form (10.51) prior to transformation, and convert them back after transformation. The brute force way of construction this mapping is to first convert the polynomial sequences 'out of' the domain of sampled polynomials (the mapping V^{-1}) and subsequently 'into' the domain of altered polynomials (the mapping W).

Lemma 10.9 (The Condition Matrix)
Define the $N \times N$ matrices

$$V_{m,n} = \int t^m \phi(t-n)dt \qquad \text{and} \qquad W_{m,n} = \int t^m \varphi_{0,n}^{\text{left}}(t)dt, \tag{10.54}$$

$m, n = 0, \ldots, N-1$. *The matrices are non-singular and* $\mathbf{A} = \mathbf{W}\mathbf{V}^{-1}$ *is an bijective mapping from the* N *dimensional vector space of sequences on the form*

$$c_n = \sum_{m=0}^{N-1} a_m n^m, \quad n = 0, \ldots, N-1 \tag{10.55}$$

to the N *dimensional vector space of sequences on the form*

$$\tilde{c}_n = \left\langle p, \varphi_{0,n}^{\text{left}} \right\rangle, \quad n = 0, \ldots, N-1, \tag{10.56}$$

where $\mathbf{a} \in \mathbb{R}^N$ *and* $p(t) = \sum_{m=0}^{N-1} b_m t^m$.

Proof
The linear independence of the row of $\mathbf{V}$ and $\mathbf{W}$ is established with equivalent arguments. Beginning with $\mathbf{W}$, first observe that if it was singular then one linear combination of the rows would be the zero column vector. The corresponding polynomial p' is then orthogonal to all the $\varphi_{0,n}^{\text{left}}$. But there exists a finite linear combination of the $\varphi_{0,n}^{\text{left}}$ and the $\phi_{0,m}$, $m \geq N$, which coincides with p' on $[0; 2N-1]$. Since p' is orthogonal to the $\varphi_{0,n}^{\text{left}}$ this combination reduces to a combination of the $\phi_{0,m}$, $m \geq N$, which vanishes identically on $[0, 1]$. Since $p \neq 0$ on $[0; 1]$ this is a contradiction. The exact same argument holds for $\mathbf{V}$ when $\varphi_{0,n}^{\text{left}}$ is replaced by $\phi_{0,m}$, $0 \leq m \leq N-1$.

Since $\mathbf{A}$ is invertible and defined on all sequences of the form (10.55) it is injective, and since the dimensions are the same, it is also surjective.

Any sequence on the form (10.55) can be written as linear combinations of the columns of $\mathbf{V}$. Thus there exists a vector $\boldsymbol{\alpha}$ such that $\mathbf{c} = \mathbf{V}\boldsymbol{\alpha}$, so

$$\mathbf{A}\mathbf{c} = \mathbf{W}\mathbf{V}^{-1}\mathbf{V}\boldsymbol{\alpha} = \mathbf{W}\boldsymbol{\alpha},$$

which is a sequence on the form (10.56). □

The mapping constructed here applies to the left edge only, and accordingly the matrix is denoted $\mathbf{A}_{\text{left}}$. It is applied to the left end of the signal prior to transformation in order convert the left end of the signal into a sequence on the form (10.56). In a similar way the mapping $\mathbf{A}_{\text{right}}$ for the right end of the signal is constructed. Applying it to the right edge prior to transformation produces the result shown in Fig. 10.3(c) on page 243, where the high pass part is indeed the zero sequence. However, the low pass part is still not a third degree polynomial. This is because signal is still in 'the domain of adulterated edges', and it is necessary to transform the signal back to 'the domain of polynomials'. This is

accomplished by postconditioning the signal by multiplying the left and right edges with $\mathbf{A}_{\text{left}}^{-1}$ and $\mathbf{A}_{\text{right}}^{-1}$, respectively. The result is shown in Fig. 10.3(d).

It is an important point that once the signal has been transformed to 'the domain of adulterated edges' one can do several consecutive transforms, say a wavelet packet decomposition, without at any step loosing the ability to map polynomials to polynomials in the low pass part and the zero sequence in the high pass part. Once the desired decomposition is determined and the corresponding decomposition signal is found, the postconditioning can be applied to retrieve the signal in 'the domain of polynomials'.

To determine $\mathbf{A}$ explicitly it is necessary to compute the all the quantities in (10.54). But computing those directly is a cumbersome task, since the scaling functions are only available numerically. Fortunately, the $\mathbf{A}$ matrix can be computed using the known one-to-one correspondence between polynomials of degree $N-1$ and the polynomial coefficient sequences of their expansion in the $\phi(t-n)$. This is demonstrated in the following lemma.

Lemma 10.10 (Numerical Construction of Conditioning Matrices)
Define $\tilde{V}_{m,n} = \binom{n}{m}$. *Then* $\mathbf{A} = \tilde{\mathbf{E}}\tilde{\mathbf{V}}^{-1}$, *where* $\tilde{\mathbf{E}}$ *is defined in Theorem 10.4.*

Proof
Note first that

$$\begin{aligned}\tilde{V}_{m,n} &= \binom{n}{m}\sum_k \int \phi(t-N+1+k)\phi(t-N+1+n)dt \\ &= \int \sum_k \binom{k}{m}\phi(t-N+1+k)\phi(t-N+1+n)dt \\ &= \int q_m(t)\phi(t-N+1+n)dt\ ,\end{aligned}$$

where the last equality defines $q_m(t)$. This implies that the corresponding mapping into the space of sequences on the form (10.56) should be

$$\tilde{W}_{m,n} = \int q_m(t)\varphi^{\text{left}}_{0,N-1-n}(t)dt = \int_0^{N-1} \varphi_m(t)\varphi^{\text{left}}_{0,N-1-n}(t)dt$$

where the second equality follows from (10.20). Thus $\tilde{\mathbf{W}}$ is the transition matrix $\tilde{\mathbf{E}}$. □

It is now easy to construct the $\mathbf{A}$ matrix necessary for the transformation.

10.5 Examples of Edge Filters

So far there has been no attempt to actually compute any edge functions or edge filter taps. This section is therefore dedicated to give some examples on both. The MATLAB code needed for making these examples are given in Appendix C. The first examples

Table 10.1: Filter taps for two Daubechies filters.

	Daubechies 4		Daubechies 8	
n	h_n	g_n	h_n	g_n
0	-0.1294	-0.4830	-0.0106	-0.2304
1	0.2241	0.8365	0.0329	0.7148
2	0.8365	-0.2241	0.0308	-0.6309
3	0.4830	-0.1294	-0.1870	-0.0280
4			-0.0280	0.1870
5			0.6309	0.0308
6			0.7148	-0.0329
7			0.2304	-0.0106

is the classical Daubechies 4 filter with two vanishing moments. The filter is given in Table 10.1. Since each moment results in four edge functions, two scaling functions and two wavelets, there is a total of eight functions, which are shown in Fig. 10.4. A number of features are apparent in these graphs. Firstly, the staggered support is immediately visible, in particular because the functions are shown on their support, i.e. the vanishing intervals are not drawn. Secondly, the functions are well-behaved in the sense that they

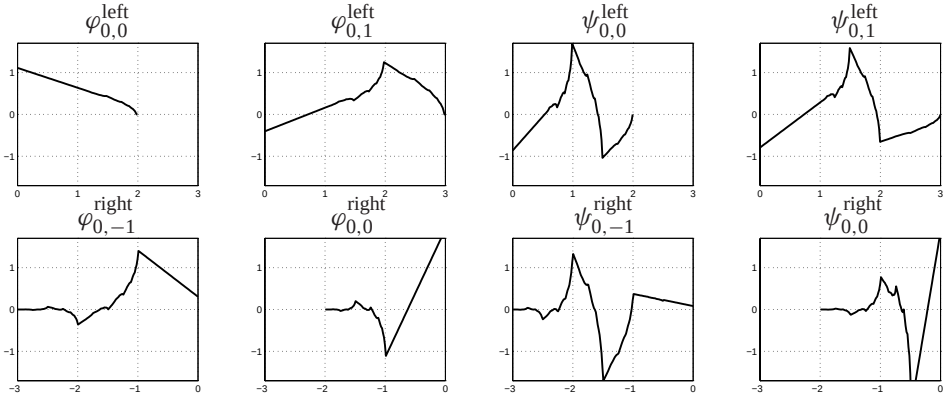


Figure 10.4: The edge scaling functions and wavelets from Daubechies 4 ($N = 2$). The functions are shown on their support, which is staggered as described in Theorem 10.4 and 10.6.

do not oscillate vigorously as is the case for the (less subtle) Meyer construction, see Cohen et al. [22, p. 64-69]. Thirdly, it is interesting to notice that on the interval $[0; 1]$ the φ_k^{left}, $k = 0, \ldots, N-1$ are pure polynomials of degree $N-1$. This is because the left most interior scaling functions was sacrificed to make room to N edge scaling functions. Consequently, the interior scaling functions have no influence for $t < 1$, and the adapted edge scaling functions are therefore polynomials on the first unit interval.

The edge filter taps for Daubechies 4 is given in Table 10.2. The filter taps come from two different equations, namely (10.33) and (10.34), for $n < N$ and $n \geq N$, respectively.

Table 10.2: The edge filters for Daubechies 4 ($N = 2$).

n	$h^{\text{left}}_{0,n}$	$h^{\text{left}}_{1,n}$	$g^{\text{left}}_{0,n}$	$g^{\text{left}}_{1,n}$
0	0.8705	-0.1942	-0.2575	-0.3717
1	0.4349	0.1902	0.8014	0.3639
2	0.2304	0.3750	-0.5398	0.7176
3		0.7676		-0.4011
4		0.4431		-0.2316

$\mathbf{A}_{\text{left}}$	
2.0963	0
-0.8008	1.0898

n	$h^{\text{right}}_{0,n}$	$h^{\text{right}}_{1,n}$	$g^{\text{right}}_{0,n}$	$g^{\text{right}}_{1,n}$
−4	-0.1292		0.4830	
−3	0.2238		-0.8366	
−2	0.8501	-0.3983	0.2274	0.2588
−1	0.4573	0.6909	0.1224	-0.5464
0	0.0375	0.6033	0.0100	0.7965

$\mathbf{A}_{\text{right}}$	
1.0014	0.0372
0	0.3249

This division is shown in the table with a gray line.

The next example is the Daubechies 8 filter, which has four vanishing moments. The characteristics discussed in the Daubechies 4 examples are equally apparent in this case. It is therefore left uncommented.

10.6 The Problem of Numerical Instability

There still exist one major problem with the particular construction laid out in the previous sections. Although a lot of effort was put into constructing the edge functions with just the right properties, the construction fell short of providing a set of orthonormal edge functions. Consequently, it was necessary to introduce the conditioning matrices. Unfortunately, these matrices turn out to be numerical unstable. In Fig. 10.6 the condition numbers (the ratio between largest and smallest singular value) of $\mathbf{A}_{\text{left}}$ and $\mathbf{A}_{\text{right}}$ are shown for the Daubechies filter with $N = 2$ through $N = 12$.

The problem traces back to $\tilde{\mathbf{V}}$ and $\tilde{\mathbf{E}}$ in Lemma 10.10, which both grows exponentially in condition number for increasing size. In some cases the matrix product $\tilde{\mathbf{E}}\tilde{\mathbf{V}}^{-1}$ do have a smaller condition number than both of the matrices, as is evident from Fig. 10.6. Note that another implementation of the $\mathbf{A}$ matrices will not solve the problem, since in this construction, they are unique.

It is interesting to note that while the condition number of the right and left condition matrices are of the same magnitude, the entries in the one matrix has is of a magnitude which is equal to the reciprocal of the magnitude of entries in the other matrix. This in some sense shift the stability problem to the one edge of the signal for the forward transform and to the other edge for the inverse transform.

To demonstrate what happens when a filter of even moderate length is applied to a non-polynomial signal, Fig. 10.7 shows the transform of a third degree polynomial followed

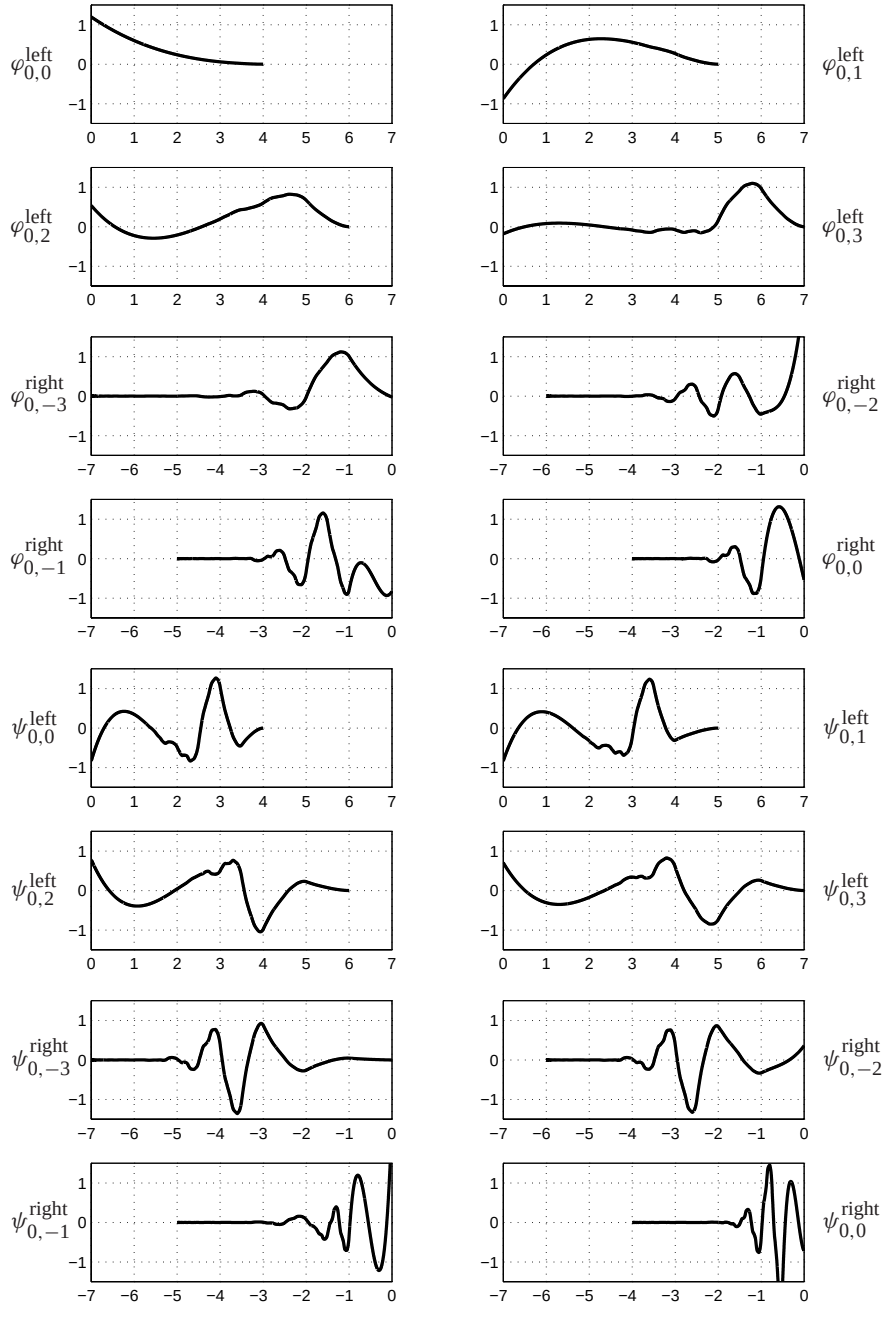


Figure 10.5: The edge scaling functions and wavelets from Daubechies 8.

Table 10.3: The edge filters for Daubechies 8 ($N = 4$).

n	$h_{0,n}^{\text{left}}$	$h_{1,n}^{\text{left}}$	$h_{2,n}^{\text{left}}$	$h_{3,n}^{\text{left}}$	$g_{0,n}^{\text{left}}$	$g_{1,n}^{\text{left}}$	$g_{2,n}^{\text{left}}$	$g_{3,n}^{\text{left}}$
0	0.9220	-0.3137	0.1308	-0.0371	-0.0185	-0.0517	0.0983	0.1427
1	0.3629	0.5401	-0.3350	0.1067	0.1585	0.3058	-0.4070	-0.4107
2	0.1268	0.5595	-0.0747	-0.0047	-0.5515	-0.5282	0.2861	0.0180
3	0.0445	0.4073	0.1410	-0.0616	0.7591	-0.1493	0.3922	0.2371
4	0.0139	0.2998	0.3020	-0.0865	-0.3070	0.7406	0.2397	0.3332
5		0.1928	0.4180	-0.0506		-0.2212	-0.7043	0.4329
6		0.0621	0.5672	-0.0940		-0.0713	-0.0828	-0.3763
7			0.4831	0.0993			0.1610	-0.5279
8			0.1557	0.6520			0.0519	-0.0091
9				0.6926				0.1799
10				0.2232				0.0580

n	$h_{0,n}^{\text{right}}$	$h_{1,n}^{\text{right}}$	$h_{2,n}^{\text{right}}$	$h_{3,n}^{\text{right}}$	$g_{0,n}^{\text{right}}$	$g_{1,n}^{\text{right}}$	$g_{2,n}^{\text{right}}$	$g_{3,n}^{\text{right}}$
−10	-0.0106				0.2304			
−9	0.0329				-0.7148			
−8	0.0312	-0.0317			0.6309	0.2283		
−7	-0.1883	0.0983			0.0279	-0.7085		
−6	-0.0272	-0.2014	-0.2017		-0.1870	0.6070	0.0500	
−5	0.6318	0.3518	0.6259		-0.0308	0.0844	-0.1551	
−4	0.7140	-0.2837	-0.5036	-0.3031	0.0328	-0.2278	0.1027	0.0250
−3	0.2292	-0.0236	-0.1757	0.9405	0.0105	-0.0638	0.1342	-0.0964
−2	-0.0029	0.6177	-0.1732	-0.0985	-0.0001	0.0865	0.7374	0.1659
−1	0.0008	-0.5014	0.3837	0.0692	0.0000	-0.0698	0.3767	0.6706
0	0.0058	0.3332	0.3253	0.0952	0.0003	0.0447	-0.5091	0.7161

$\mathbf{A}_{\text{left}}$				$\mathbf{A}_{\text{right}}$			
16.588				1.0000	0.0127	-0.0087	-0.0000
-34.605	3.7077				0.3346	-0.4869	0.5882
26.731	-3.3528	1.4796				0.0525	-0.1891
-7.270	1.0165	-0.2701	1.0321				0.0350

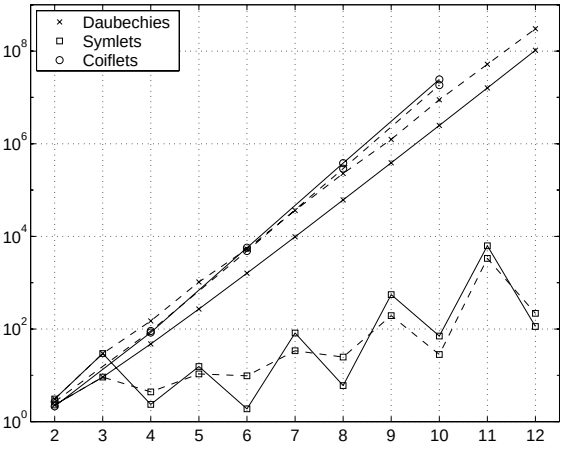


Figure 10.6: The condition numbers for $\mathbf{A}_{\text{left}}$ (solid) and $\mathbf{A}_{\text{right}}$ (dashed) for the three types of filters. The horizontal axis is the number of vanishing moments, the vertical axis is the magnitude of the condition numbers.

by a transform of the same polynomial with mild noise added.

10.6.1 Obtaining Numerical Stability

The numerical instability of the conditioned transform, as described previously in this section, is so severe that it cannot simply be ignore or accepted. This is clear demonstrated in Fig. 10.6 and 10.7. At the same time the polynomial regenerating filters seems too promising to be abandoned as edge handling method. Thus, the question addressed in this section is how to modify the method in order to achieve numerical stability without sacrificing the polynomial regenerating property.

It was noted in the previous section that condition matrices $\mathbf{A}_{\text{left}}$ and $\mathbf{A}_{\text{right}}$ are unique for a given set of edge scaling functions. Consequently, these cannot be chosen differently. However, the condition matrices are, according to Lemma 10.10, defined by $\mathbf{A} = \tilde{\mathbf{E}}\tilde{\mathbf{V}}^{-1}$, where $\tilde{\mathbf{E}}$ is a non-unique orthonormalization matrix. One choice of $\tilde{\mathbf{E}}$ is given in the constructive proof on Theorem 10.4, but there are many other choices, since multiplying $\tilde{\mathbf{E}}$ with a orthogonal transform will give another orthonormalizing matrix. But even this freedom is not enough to solve the problem.

Lemma 10.11
There exists an orthonormal set $\{f_k\}_{k=0,\ldots,N-1}$ *obtained by an orthonormalization of the*

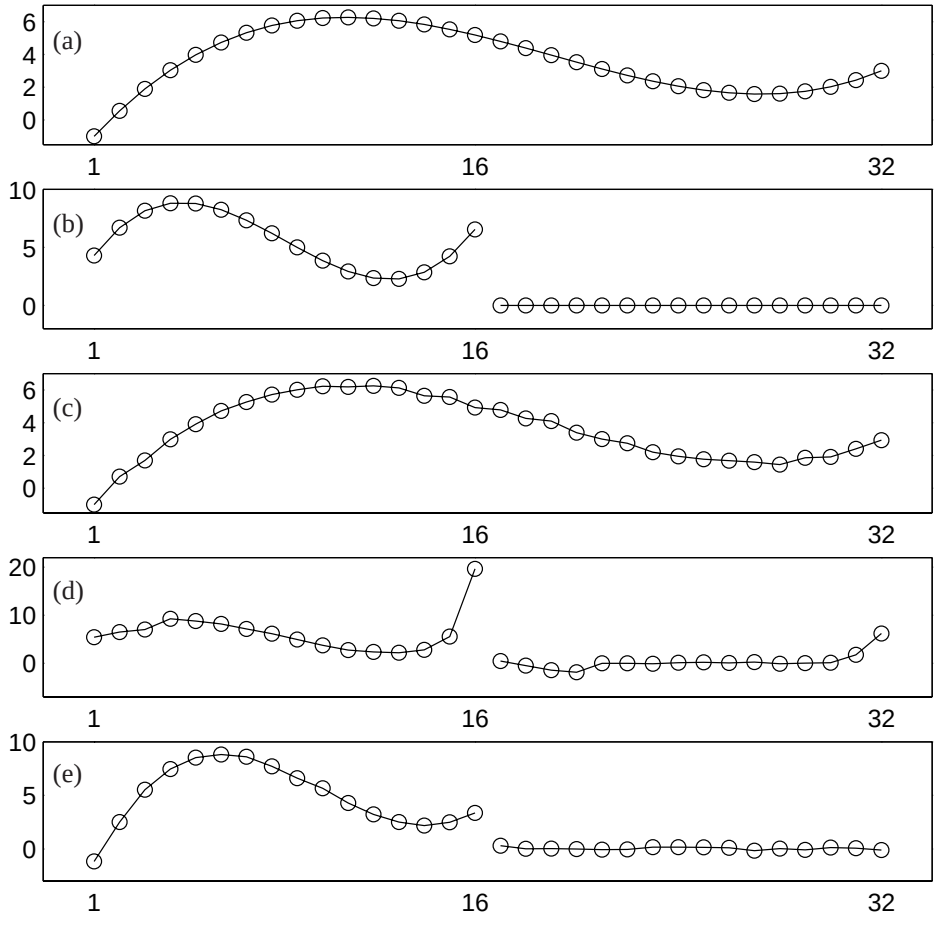


Figure 10.7: In (a) the polynomial $t^3 - 4t^2 + 2t + 6$ is sampled in 32 points in the interval $[-1; 3]$, and the result of preconditioning, transformation with edge filters, and postconditioning is showed in (b). Daubechies length 8 filters are used. This produces the expected result: The high pass part is completely vanishing, while the low pass part is another polynomial sampled in 16 points. When even mild normal noise is added to the polynomial, (c), the result of transforming using pre- and postconditioning is highly numerical unstable, as is clearly seen in (d). Using Symlets length 8 filter instead, (e), produces a more stable result (which is hinted in Fig. 10.6 at $N = 4$). Note that (b) and (e) are not directly comparable since two different filters are used.

set $\{\varphi_k\}_{k=0,\ldots,N-1}$ *such that* $\int f_k(t)dt = 1$ *if and only if*

$$\sum_{k=0}^{N-1}\left|\int \varphi_k^{\text{left}}(t)dt\right|^2 = N \ . \tag{10.57}$$

The link between φ and φ^{left} is given in Theorem 10.4 on p. 236.

Proof
Assume first that the set $\{f_k\}$ exists. This set is orthonormal if and only if there exists a orthogonal matrix $\mathbf{B} \equiv [\beta_{m,n}]$ such that

$$\begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{bmatrix} = \mathbf{B} \begin{bmatrix} \varphi_0^{\text{left}} \\ \varphi_1^{\text{left}} \\ \vdots \\ \varphi_{N-1}^{\text{left}} \end{bmatrix}$$

since

$$\langle f_s,\, f_u\rangle = \left\langle \sum_{m=0}^{N-1}\beta_{s,m}\varphi_m^{\text{left}}, \sum_{n=0}^{N-1}\beta_{u,n}\varphi_n^{\text{left}} \right\rangle = \sum_{m=0}^{N-1}\beta_{s,m}\beta_{u,m} = \delta_{s,u} \ ,$$

where the last equality is valid for a orthogonal $\mathbf{B}$ only. Then

$$\begin{aligned} N &= \sum_{m=0}^{N-1}\left|\int f_m(t)dt\right|^2 \\ &= \sum_{m=0}^{N-1}\left|\sum_{n=0}^{N-1}\beta_{m,n}\int \varphi_n^{\text{left}}(t)dt\right|^2 \\ &\le \sum_{n=0}^{N-1}\sum_{m=0}^{N-1}\left|\beta_{m,n}\right|^2\left|\int \varphi_n^{\text{left}}(t)dt\right|^2 \\ &= \sum_{n=0}^{N-1}\left|\int \varphi_n^{\text{left}}(t)dt\right|^2 \end{aligned}$$

together with

$$\sum_{m=0}^{N-1}\left|\int \varphi_m^{\text{left}}(t)dt\right|^2 \le \sum_{m=0}^{N-1}\int\left|\varphi_m^{\text{left}}(t)\right|^2 dt = N$$

gives (10.57).

The other way is not proved here, as the author by the deadline of the thesis did not have a finished proof. □

The Daubechies 4 and 8 filters do not satisfy (10.57), and in fact, neither does any of the

Daubechies filters. Consequently, it is not possible in the case of these filters to achieve unit integral by means of a different orthonormalization matrix $\tilde{\mathbf{E}}$ in Theorem 10.4. This also means that giving up the staggered support will not solve the problem.

This leaves two choice: Either an ad hoc solution is used to reduce the effect of the conditioning matrices, or a theoretically founded solution is introduced by changing the construction at a very early stage. The former choice is obviously dependent on the signal processing task at hand. In particular, it is important whether perfect reconstruction and the frequency interpretation is needed. The latter choice, although it requires a good idea and a lot of work, is clearly preferable to the former.

One ad hoc solution is to simply disregard the conditioning matrices. It is not a very attractive solution, however. This is evident from the plot in Fig. 10.3(b). The deviation from the expected result is evident, in particular the coefficients at the right end of the signal differ significantly from a third degree polynomial and the zero signal, respectively.

Another solution is to alter the signal at the ends prior to transformation, for instance by replacing the outer most coefficients by a (sampled) low degree polynomial fitted to the original coefficients. This will only solve the problem if at least $2^{J-1}(3N-3)$ samples are replaced in the one end (the end where the large condition number of the preconditioning matrix is due to medium to very small entries), where J are the number of consecutive transform steps, i.e. levels in the decomposition, and N samples are replaced at the other end (the end where the large condition number is due to medium to very large entries). The former number can quickly become large compared to the length of the signal. The $2^{J-1}(3N-3)$ is motivated by the fact that the longest edge filter in this construction has $3N-1$ filter taps, and the effect kicks in with the second edge filter (it is currently unknown why the first edge filter does not cause instability). While this method does produce good results it is an 'unnatural' way of handling the problem. The numerical instability is still latent in the conditioning matrices, and implementation in low-cost signal processing hardware becomes difficult (if not impossible). It also requires significantly more computations since an approximating polynomial has to be determined prior to transformation.

A more theoretical approach is to introduce more freedom in the construction, and thereby allowing for an orthogonalization of the scaling functions such that unit energy is achieved. More freedom is available by claiming more than the just the first interior scaling function (which was claimed at the end of Section 10.3.1). This approach was suggested to the author by Jan Olov Strömberg. But although it seems promising the author still has not found the time to investigate it any further.

10.7 Application of Edge Filters to Real Measurements

The presentation and discussion of moment preserving edge filters has so far been theoretical. The construction of the edge filters and the conditioning matrices have been presented in detail with the corresponding MATLAB code. The filters have been shown to possess a series of useful properties. But the construction also has a significant stability

problem. To demonstrate that the filters can indeed be used as long as the right wavelet filter is chosen this section applies the method to a real signal with low frequency noise. For comparison the Gram-Schmidt orthogonalized edge filters, see Section 9.4, are also applied.

In Fig. 10.8(a) an example of a very powerful low frequency noise is shown. This noise has been generated moving a neon tube, mounted with a metal protection grid, in the vicinity of the receiver. The shown signal contains 3072 samples and has been recorded at 5.7 kHz. The shown signals thus represents 0.54 seconds. The receiver used for making the recording is the same as is used in the measurement of reflection maps, see 8.3.1. The noise contains a 100 Hz disturbance from the oscillations of the light, and a somewhat slower oscillation. The latter is caused by the movement of the light which in turn causes the metal grid to occasionally cover the neon tube partially.

The following examples is presented to demonstrate what happens when a sensor employing the wavelet modulation is subjected to this type of disturbance. The entire process is much the same as in the first test setup, see Section 5.2. A five level WPT of a length 512 sample signal is used, and the original signal is designed such that the 6th of 16 elements are non-vanishing. This element is the 19th RS sequence (row) of a 32×32 RST matrix. The reason for using a SS sequence here is explained in Section 4.7.2. This designed signal is inversely WP transformed (using the Symlets 12 taps filter), and the transmission is simulated by scaling this signal and adding the noise. The interval [1024; 1536] in Fig. 10.8(a) is chosen as the noise, and the simulated, received signal is shown in Fig. 10.8(b). The transmitted signal is mixed with the white noise in the signal and is therefore not explicitly visible in the plot.

The plots in Fig. 10.8(c) and (d) show the result of using a set of Gram-Schmidt orthogonalized edge filters and a set of moment preserving edge filters with conditioning matrices to generated and post-process the signal. Note that the DC component is present in the first element which is therefore way off the scale. The artifacts generated by the introduced discontinuities in the GS edge filter case are clearly visible in Fig. 10.8(c). It is also easy to see the benefit of using the moment preserving edge filters as there are (almost) no artifacts in Fig. 10.8(d).

The designed signal is vanishing except for the 6th element which corresponds to the interval [160; 192]. In the 6th element of the GS edge filter WPT the edge effect is most noticeable at the left edge. Actually, there seems to be no effect at the right edge. The RST of the 6th element does not reveal the 19th sample as significant, see Fig. 10.8(e). This is mainly because the the first sample of the 6th element contains most of the energy in the 32 samples. Thus, the RST shown in Fig. 10.8(e) is mostly the first RS sequence. The amplitudes of these edge effects in Fig. 10.8(c) are not fixed, but change throughout the elements with the low frequency noise. Or more accurately, with the different between the amplitudes at the ends of the signal (since the edge effects are artifacts generated by the discontinuity in the periodized signal). In fact, they almost disappear when the two edges happen to ‘meet’. But only almost since a discontinuity in the first derivative of the signal also contributes a little to the edge artifacts.

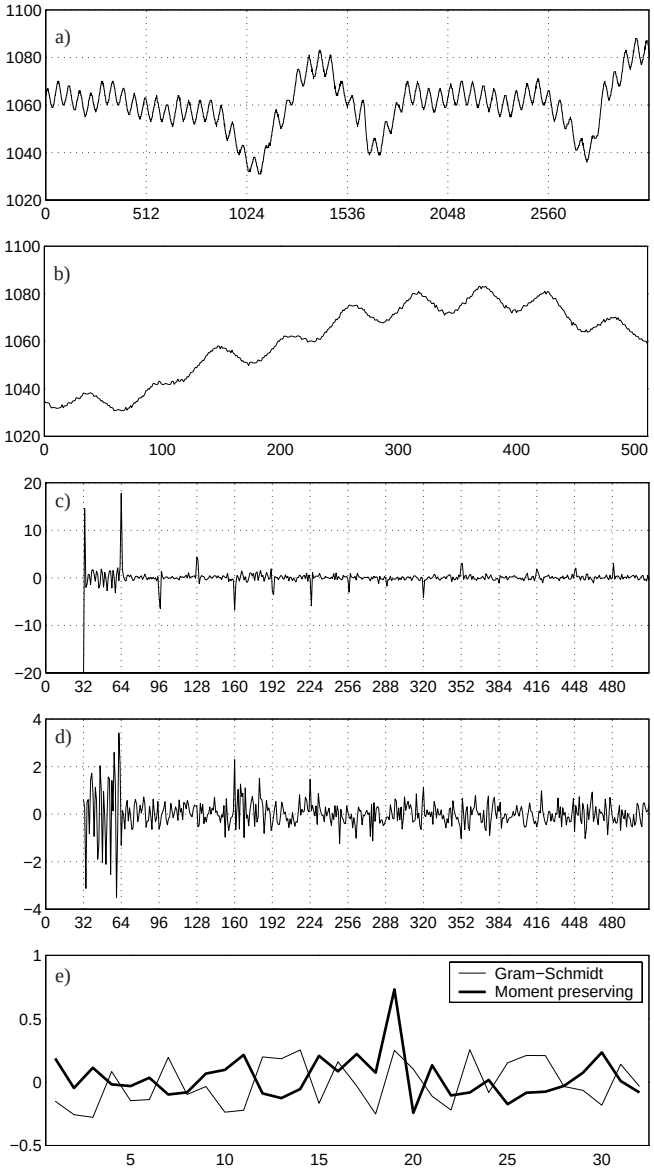


Figure 10.8: The effect of using Gram-Schmidt and moment preserving edge filters on real signal. (a) an example of low frequency noise. (b) the transmission signal plus the noise in [1024; 1536]. (c) WPT of transmitted signal, with Gram-Schmidt edge filters, (d) WPT with moment preserving edge filters, (e) RST of the 6th element. See text for further explanations.

When the moment preserving edge filters have been used the edge artifacts are avoided and the RST of the 6th element now shows a significant difference between the 19th sample and the other samples, see Fig. 10.8(e). Note that the two signals in Fig. 10.8(e) has been energy normalized for easy comparison.

At this point one may wonder why this application of the moment preserving edge filters went well despite the fact that the transform is numerical unstable. The reason is that the Symlets 12 taps filter is one of the few relatively stable filters (see Fig. 10.7 and recall that Symlets 12 has 6 vanishing moments). Nonetheless, the are some small artifacts in the transformed signal. This is due to the fact that the transform is indeed not entirely stable. The condition number of the condition matrices is approximately 10 which does cause some minor instability, as is seen in Fig. 10.8(d).

10.8 Conclusion

It is pointed out in Depczynski et al. [29] that the idea of using an existing wavelets basis, where the interior functions are maintained while the edge functions are altered involves a "complicated reorthogonalization process, with an impact on the organization and the 'complexity' of fast wavelet algorithms". They suggest a construction where the original wavelet basis is constructed especially for bounded intervals. While this author tends to disagree with the first observation (this chapter does present the filter taps needed in a fast implementation, and so does the original paper by Cohen et al.) the idea of specially constructed wavelets seems appealing. An example of this is given in Depczynski [28].

Still, the edge filters constructed in this chapter have a series of useful properties which are worth fighting for. This is evidenced by the fact that more than 150 papers from all sorts of proceeding and journals have cited the paper by Cohen et al. [22] (result of search on ISI Web of Science). However, browsing through these papers did not reveal a single one pointing out the numerical instability reported in this chapter.

A couple of suggestions for handling the instability problem was given. However, the constructive ones do not seem to be particularly useful, and the more theoretical approach is still just a sketchy idea. It should be noted, though, that if one is satisfied with the length 12 or 16 Symlets filters the instability is at a reasonable level according to Fig. 10.6. MATLAB code for generating the edge filters and conditioning matrices is given in Appendix C. This code reproduces exactly the derivations in Section 10.3 and 10.4.

There exists two alternatives to the orthogonal filter bank implementation of the wavelet transform which is the basis for all the edge handling methods presented in this and the previous chapter. The most obvious, perhaps, is relaxing the orthogonality constraint and use biorthogonal wavelets. While the energy preserving property is lost a number of useful properties are gained. Among those is more freedom to design methods for the edges of the signal. This is the approach that Cohen has suggested to the author. Another alternative is to use the lifting technique for designing the wavelet transform and thus the action taken in the end of the signal. A description of this is found in Jensen and la Cour-Harbo [45].

Despite the numerical stability issue it is possible to apply the moment preserving edge filters successfully. This was demonstrate in Section 10.7. The Symlets 12 tap filter was used because it produces relatively stable edge filters and conditioning matrices. The result is a more accurate estimate of the CGM when high power low frequency noise is present in the signal.

11 The Rudin-Shapiro Transform

This chapter presents the Rudin-Shapiro transform in a mathematical and historical context. This linear transform was originally conceived as a series of coefficient sequences from a set of trigonometric polynomials, but it now exists in is own right. The transform has a series of nice properties among which the spread spectrum property of the basis elements is the most noticeable one. The transform proves useful for designing signals in low-cost hardware, not least due to the existence of a fast and numerically robust implementation.

The aforementioned polynomials are often categorized as flat polynomials. This refers to the fact that the amplitude of the polynomials are bounded by a constant times the energy of the polynomial. There exists many other types of flat polynomial than just the Rudin-Shapiro polynomials, and the history of the development in the field of flat polynomials is quite interesting (at least to the author of this thesis). This is in no small part due to the fact that a number of seemingly simple questions within the field have remained unanswered for several decades.

It was demonstrated in the chapters in Part I that spread spectrum transforms have a role to play in the attempt to increase the robustness of active sensors. The aim of this chapter is therefore also to introduce the signal processing aspects of the Rudin-Shapiro transform. In particular, in is interesting to identify a series properties which are useful in the design of transmission signals.

11.1 Search for Flat Polynomials

The construction of flat polynomials dates back to the beginning of 20th century. Of course, at that time the purpose was not to design signals for use in digital transmission systems. The incitement was rather a mathematical interest in certain ‘nice’ trigonometric series. In 1916 Hardy and Littlewood discovered a series, which they investigated as part of a study of so-called elliptic Theta-functions. One of their results was a series which has a property that today is referred to as (semi-)flatness. Since then many other polynomials with various similar properties were discovered.

The interest in flat polynomials still exists today, though the interest is now in general fueled by the need for pseudo random sequences suitable for application in fields such

as transmission and encryption. It is therefore research in information theory rather than pure mathematics that produces new results in the field of flat polynomials, and although many interesting results have emerged this thesis is focused on the use of one particular type of flat polynomials, namely the Rudin-Shapiro polynomials.

11.1.1 Introduction

To fully appreciate the theory presented in this chapter some basic concepts and notation is necessary. They are introduced in the following two subsections. A brief overview of the history of flat polynomials is then given in Section 11.1.4. This provides view of the various, positive and negative, results obtained in the area of flat polynomials. The authors fascination of the history of flat polynomials aside this provides a perspective view on the RS polynomials. That is, the historical overview gives an idea of what type of improvements are possible. It also demonstrates why this field of research has a gain to pain ratio close to zero.

The Rudin-Shapiro polynomials are one of many possible sets of flat polynomials. The historical overview includes all kinds of flat polynomials, some of which are rather useful in real applications.

A construction of Coifman et al. [23] is based on the idea of generating sequences which are uncompressible by a Haar-Walsh wavelet packet transform, i.e. the transform coefficients exhibits no decay. The result is sequences of ± 1 and $\pm i$ which have the same type of flatness as Rudin-Shapiro sequences.

In applications it is often very useful to have spread spectrum sequences with a good autocorrelation, i.e. where only the zero lag is significantly different from zero. Such sequences have been systematically constructed by No et al. [60, 61, 62].

11.1.2 Notation

Before venturing into a search for flat polynomials it is convenient to fix the notation. First unimodular sequences are defined. They will become the coefficients in the flat polynomials.

Definition 11.1 (Unimodular sequences)
Define the sets of unimodular sequences as

$$\mathscr{S}_N^{\,p} = \left\{ \boldsymbol{\beta} \in \mathbb{C}^N \;\middle|\; \beta_k \in \{e^{i2\pi m/p}\}_{m=0,\ldots,p-1} \right\} \quad \text{for } p = 2, 3, \ldots,$$

which means the set of N dimensional vectors with entries in a set of equidistantly sampled points on the unit circle in $\mathbb{C}$. Define also the natural extension

$$\mathscr{S}_N^{\,\infty} = \left\{ \boldsymbol{\beta} \in \mathbb{C}^N \;\middle|\; \beta_k \in \{e^{i2\pi\alpha_k}\}_{\alpha_k \in [0;1)} \right\}$$

for $p = \infty$.

The polynomials are defined on the unit circle in the complex plane, and takes coefficients from the set of unimodular sequences. Note how the defined polynomials are the Fourier transform of the unimodular sequences.

Definition 11.2 (Trigonometric Polynomials)
Define the sets of complex trigonometric polynomials

$$\mathscr{H}_N^p = \left\{ f_N : \mathbb{R} \mapsto \mathbb{C} \;\middle|\; f_N = \sum_{k=0}^{N-1} \beta_k e^{i2\pi k\xi}, \quad \boldsymbol{\beta} \in \mathscr{S}_N^p, \quad \xi \in [0;1) \right\}$$

for $p = 2, 3, \ldots, \infty$*. Define also*

$$\mathscr{H}^p = \bigcup_{n=1}^{\infty} \mathscr{H}_n^p.$$

Remark 11.2.1

- *In most literature, including the conjectures of Littlewood [54], only the two sets* $\mathscr{H}^2$ *and* $\mathscr{H}^\infty$ *are mentioned, and they are typically referred to as* $\mathscr{F}$ *and* $\mathscr{G}$*.*
- *The set* $\mathscr{H}^2$ *differs from the rest in being the only one with exclusively real coefficients* (± 1*'s). This makes it by far the most interesting set from an applicational point of view.*
- *The Rudin-Shapiro polynomials are examples of* $\mathscr{H}^2$ *functions.*
- *It is only a matter of taste whether the lower and upper bound on the sum should be* 0 *and* $n-1$*, respectively,* 0 *and* n*, or* 1 *and* n*. There seems to be no preference in the existing literature, and here the bounds are chosen to correspond with the general notion that, as default, the first index in a vector is* 0*, and that dimension of the sequence spaces (to which* β *belongs) should correspond to "dimension" of the function spaces* $\mathscr{H}^p$*.*

It is surprising that the set $\mathscr{H}^p$, which is simply a collection of Fourier transformed sequences taken from the unit circle, has been subject to extensive investigations throughout the past 50 years, and that some seemingly simple questions still remains unanswered. The Fourier transform is arguable the best understood and most popular tool in harmonic analysis, and thus one is inclined to believe that a set such as $\mathscr{H}^p$ would be well-described by now.

11.1.3 Flatness of Polynomials

The search for flat polynomials is basically a search for an answer to the question: How close can a function $f_N \in \mathscr{H}^p$ come to satisfying $|f_N| = \sqrt{N}$ for arbitrarily large N?

The question is quite intriguing because on the one hand the equality is never reached for finite N. This can be seen by first regarding the following lemma.

Lemma 11.3
Let $P \in \mathscr{H}_n^p$, $p > 1$. Then $\|P\|_\infty \geq \sqrt{N}$. The equality holds iff $|P(\xi)| = \sqrt{N}$.

Proof
The lemma follows immediately from $\|P\|_2 = \|\mathbf{c}\|_2 = \sqrt{N}$, $\mathbf{c}$ being the Fourier coefficients of P, and the fact that $\|P\|_2 \leq \|P\|_\infty$ on the unit interval. □

Assuming now that $|P(\xi)| = \sqrt{N}$ then, for $|\beta_k| = 1$,

$$N = |P(\xi)|^2 = \Big|\sum_{m=0}^{N-1} \beta_m e^{i2\pi m\xi}\Big|^2 = \sum_{m=-N+1}^{N-1} (\boldsymbol{\beta} * \overline{\boldsymbol{\beta}})_m e^{im\xi} \quad \Rightarrow$$

$$(\boldsymbol{\beta} * \overline{\boldsymbol{\beta}})_m = \delta[m] \quad \Rightarrow \quad \beta_0\beta_{N-1} = 0,$$

which is a contradiction. On the other hand, the Rudin-Shapiro polynomials introduced in Section 11.2 demonstrate that for $\mathscr{H}^2$ (and indeed for $\mathscr{H}^{2p}$ and $\mathscr{H}^\infty$) there is a uniform upper bound for the deviation of $|f_N|$ from $\sqrt{N}$. From (11.8) it is seen that this bound is $\sqrt{2}$, since $|P_n(\xi)| \leq \sqrt{2}\sqrt{2^n}$.

The question of how close a function $f_N \in \mathscr{H}^p$ can come to $\sqrt{N}$ might also involve a lower bound. Moreover, there may even exist polynomials such that $f_N(\xi)/\sqrt{N} \to 1$ uniformly in ξ for $N \to \infty$. The latter would certainly qualify as a flat polynomial. In the course of this chapter it becomes necessary to distinguish between four different types of flatness.

Definition 11.4 (Flat Polynomials)
Define for a function $f_n \in \mathscr{H}^p$ the following terms associated with the given inequalities.

Flatness	*Condition*
Semi-flat	$\|f_N\| \leq B\sqrt{N}$
Near-flat	$0 < \|f_N\| < B\sqrt{N}$
Flat	$A\sqrt{N} \leq \|f_N\| \leq B\sqrt{N}$
Ultra-flat	$(1 - \|o(1)\|)\sqrt{N} \leq \|f_N\| \leq (1 + \|o(1)\|)\sqrt{N}$

The constants A and B are independent of N.

In many scenarios, particularly in real applications, this distinction is less important as even the semi-flat polynomials exhibits spread spectrum properties (at least for reasonably small B). The discussion of the properties of the trigonometric polynomials in $\mathscr{H}^p$ in respect to different types of flatness is thus of a more academical nature.

To aid in the analysis of sequences it is necessary to have a measure of the flatness of a sequence. Typically, the design of a sequence takes place in the time domain, while the flatness is measured in the frequency domain. An obvious choice for flatness of a function is the ratio of maximum modulo and the size of the area under the function, that is the sup norm over the L^1 norm. However, the function f_N in this context is given as

the continuous Fourier transform of a finite sequence, and this sequence thus becomes the coefficients two a linear combination of elements in an orthogonal set. Hence a more apparent choice would be to use the L^∞ norm divided by the L^2 norm. While the L^2 norm as opposed to the L^∞ norm is dependent on the length of the interval on which the function to be measured is defined, a further requirement to the flatness measure is that the function is defined on a unit interval (or, alternatively, that the measure is normalized with respect the length of the interval). This leads to the crest factor, in some literature known as peak-to-mean ratio or peak-to-mean power envelope ratio.

Definition 11.5 (The Crest Factor)
For any sequence $\mathbf{c} \in \mathbb{C}^N$ *define the polynomial*

$$P(\xi) = \sum_{n=0}^{N-1} c_n e^{i2\pi n\xi}, \quad \xi \in [0; 1).$$

The crest factor $\mathcal{C}$ *for any sequence* $\mathbf{c} \in \mathbb{C}^N$ *is defined as*

$$\mathcal{C}(\mathbf{c}) \equiv \frac{\|P\|_\infty}{\|P\|_2}.$$

Note that since

$$\|P\|_2^2 = \Big\| \sum_{n=k}^{N+k-1} c_n e^{i2\pi n\xi} \Big\|_2^2 = \int_0^1 \sum_{n=k}^{N+k-1} |c_n e^{i2\pi n\xi}|^2 d\xi = \sum_{n=k}^{N+k-1} |c_n|^2 = \|\mathbf{c}\|_2^2 \quad (11.1)$$

the crest factor is also given as $\mathcal{C}(\mathbf{c}) = \|P\|_\infty / \|\mathbf{c}\|_2$. Since the crest factor quantifies the amplitude of the Fourier transform of $\mathbf{c}$ it is an indicator for the 'frequency flatness' or 'frequency spreading' of the sequence $\mathbf{c}$.

Before turning to the applicational aspects of flat polynomials, which in this thesis means the Rudin-Shapiro polynomials and sequences, the author would like to give a short historical presentation of the quest for flat polynomials.

11.1.4 A Brief Review of the History of Flat Polynomials

Many people have contributed to the development of flat polynomials, and many papers have been written on the subject. Some publications are hard to come by, either because their date back many decades, or because they are local journals of university, academies, and the like. Consequently, this presentation is not exhaustive and serves only as background information for interested readers. A summary is found in Table 11.1. Thanks are due to the library at Department of Mathematics at KTH, Stockholm, for assistance in locating some of the papers referred in here.

The fundamental question which is the incitement for virtually all of the people in the field of flat polynomials are: How close can a function $f_N \in \mathscr{H}^p$ come to satisfying $|f_N| = \sqrt{N}$ for arbitrarily large N?

		There exist polynomials of type				
		Semiflat		Flat		Ultraflat
in the set $\mathscr{H}^\infty$	$\subset [0;1)$					Littlewood, 1966 [54] Byrnes, 1977 [12]
	$= [0;1)$	Hardy, Littlewood, 1916 [38] Littlewood, 1966 [54] Beller, 1971 [5]	> 3 1.36 1.1717	Clunie 1959[1] [21] Körner, 1980 [50]	– –	Cj. Ø Erdös, 1957 [31] Kahane, 1980 [46]
in the set $\mathscr{H}^p$	$\subset [0;1)$			Beck, 1990 ($p=3$)[2] [4]	–	Beck, 1990 ($p=3$)[3] [4]
	$= [0;1)$	Coifman et. al., 1994 ($p=4$)[4] [23]	$\sqrt{2}$	Beck, 1990 ($p=400$) [4]	–	Ø : Fredman et al. 1989)[5] [32]
in the set $\mathscr{H}^2$	$\subset [0;1)$	Cj. nearflat, la Cour-Harbo, 2000	$\sqrt{2}$			
	$= [0;1)$	Shapiro, Rudin, 1951/59[6] [66] Shapiro, Rudin, 1951/59[7] [66] Brillhart, Morton, 1978[7] [9] France, Tenenbaum, 1981[7] [57] Saffari, 1986[7] [67] Allouche, France, 1985[8] [2]	$\sqrt{2}$ $2+2\sqrt{2}$ $\geq \sqrt{6}$ $2+\sqrt{2}$ $(2+\sqrt{2})\sqrt{\frac{3}{5}}$ $2+\sqrt{2}$			Cj. Ø Erdös, 1957 [31] Cj. Ø Saffari, Smith, 1990 [68]

[1]Coefficients in the unit disc.
[2]For half the unit circle.
[3]For sufficiently small neighborhood of 0.
[4]For signal length 2^j
[5]In L^4 norm and for coefficients satisfying $\beta = \overline{\beta}$
[6]Rudin-Shapiro sequences.
[7]All partial Rudin-Shapiro sequences.
[8]All 2-multiplicative sequences.

Table 11.1: The results obtained so far in the search for flat polynomials.

One of the first clues was given in 1916 by Hardy and Littlewood [38], who studied the series

$$\sum_{n=1}^{\infty} e^{ikn\log n}\frac{e^{in\xi}}{n^{1/2+\alpha}}, \quad c,\alpha \neq 0. \tag{11.2}$$

When $\alpha = -1/2$ the partial sum $|s_N(\xi)|$ is uniformly bounded by $C\sqrt{N}$ on $[0;2\pi]$ with C depending only on k (Zygmund [85]) making the series a semi-flat polynomial. No explicit bound is given in the book, but a few numerical experiments reveals that $C > 3\sqrt{2\pi}$ for $k = 1$. This means that 3 is a the lower bound for the crest factor of the sequence $c_n = e^{in\log n}$. The polynomials is shown in Fig. 11.1, which incidentally it below 3. This is due to the resolution of the calculations and the graph. Zooming in on the third top reveals that it, with sufficiently many terms of the sum, does reach above 3.

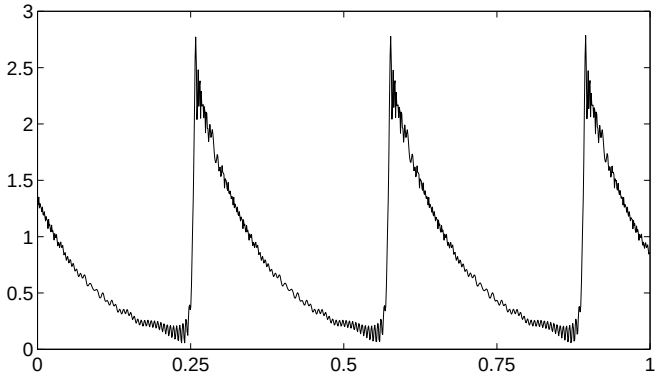


Figure 11.1: The polynomial (11.2) with $\alpha = -1/2$ and $k = 1$, here shown with the first 1000 terms of the sum. The coefficients are normalized to have norm 1.

In 1957 Paul Erdös gave presented at a symposium at Assumption University of Windsor a list of 28 so far unsolved problems [31]. Number 22 reads: If $f_N \in \mathscr{H}^\infty$, does there exist a universal constant $c > 0$ such that $\|f_N\|_\infty > (1+c)\sqrt{N}$? This is the opposite of conjecturing than there exists ultra-flat polynomials $f_N \in \mathscr{H}^\infty$. The existence of such polynomials was confirmed in 1980 by Kahane [46]. And in 1989 Fredman et al. [32] proved that $\|f_N\|_4 > 1.1048^{1/4}\sqrt{N}$ when $\boldsymbol{\beta} = \bar{\boldsymbol{\beta}}$. Erdös claimed that he had an unpublished proof that

$$\left\|\sum_{k=0}^{N}\beta_k\cos k\theta\right\|_\infty > (1+c)\sqrt{N/2}\,,$$

which is a variation on the theme. He did not reveal the value of the constant c, though. He also mentioned, as problem number 26, the question of whether there exists a flat $f_N \in \mathscr{H}^2$.

Prior to this Golay had in 1949 in a paper titled 'Multislit spectrometry' [33] introduced the notion of pairs of complementary series. Although the definition from then does not immediately reveal it, complementary series are coefficients in flat polynomials. The theory was further develop in 1962 [34]. Since then others have further refined the theory to include whole classes of complementary series and to include multiphase series instead of just ± 1's.

In the mean time, the same idea was discovered by mathematicians and formed a independent line of investigation. Harold Shapiro had studied extremal problems of trigonometric series in his Master's thesis from 1951 [70], and from this derived examples of complementary series (although he obviously does not refer to them by this name). On page 39 the definition of Rudin-Shapiro polynomials (11.3), (11.4) is given, and the crest factors $\sqrt{2}$ for length 2^n and $2+\sqrt{2}$ for arbitrary length are deduced. These results were rediscovered in 1959 by Rudin who, with the accept of Shapiro published the paper 'Some theorems on Fourier coefficients' which introduced the construction as it is shown in the next section.

While the engineers who took an interest in flat polynomials were looking for binary sequences with nice autocorrelation properties, the interest on the mathematicians part was in peak values of polynomials defined with a set of restrictions. These typically included unimodular coefficients and restriction to the unit circle in $\mathbb{C}$. Many other restrictions have been applied, probably due to the difficulty in achieving any significant results.

In 1965 Newman [59] investigated the problem of creating a truly flat polynomial in L^1 norm. He presents a certain construction which yields flat polynomials in L^1 as well as in L^4. The same challenge was also taken up by Littlewood in 1962 [53], though he attempted the construction in L^2 norm. He showed that the function

$$\sum_{m=0}^{N-1} \exp\Big(\frac{1}{2}m(m+1)\theta\pi i/N\Big)$$

tends to 1 uniformly for $N \to \infty$ on $N^{-1/2+\delta} \leq |\theta| \leq \pi$ (but fails outside this interval). Littlewood states explicitly that he has made extensive attempts to modify the construction to achieve uniform convergence for all θ.

The extremal problems in L^p, $p > 2$, is investigated in 1971 by Beller [5], and certain polynomials are shown to converge to 1. However, the sup norm is asymptotic to $1.1716\ldots$, and thus does not qualify the polynomials are truly flat.

In 1980 Körner [50], using a construction by Byrnes [12], proved that there exists flat polynomials $f_N \in \mathscr{H}^\infty$. Soon after Kahane significantly improved this by disproving problem number 22 by Erdös and thus showing the existence of ultra-flat polynomials. This is one of the major result in the field of flat polynomials.

The existence of ultra-flat polynomials with real, unimodular coefficients have been very difficult to settle. A number of mathematicians have actually published works proving as well as disproving the existence. The author of this thesis have not been able to determine whether the question has indeed been settle definitively.

11.2 Classical Rudin-Shapiro Polynomials

The first discovery of systematic construction of sequences which is somewhat flat in the frequency domain was done by Golay in 1949 [33]. He introduced the notion of complementary series. A set of complementary series is defined as a pair of equally long, finite sequences of $+1$'s and -1's such that the sum of the autocorrelation coefficients of the two sequences is zero for even shifts except for the zero shift. Later he further developed the theory of such pairs, see Golay [34], showing the one set of series could produce several others.

The idea of complementary series was discovered independently by Shapiro in his 1951 Master's thesis [70]. According to Shapiro, he 'accidentally' made the discovery as he was working on extremal problems for polynomials. He thus had a mathematical approach to the subject whereas Golay took a more engineering approach. The Shapiro result was rediscovered by Rudin and published in 1959 [66], and is now known as the Rudin-Shapiro polynomials. The construction is recursive and generates a pair of (semi-)flat polynomials for each power of 2 Actually, the coefficients in these polynomials is the very same as the binary Golay complementary series. This is easily verified once the Rudin-Shapiro polynomials have been defined, see Section 11.2.2.

11.2.1 Rudin-Shapiro Polynomials

The Rudin-Shapiro polynomials are defined recursively as

$$P_{n+1}(\xi) = P_n(\xi) + e^{i2\pi 2^n \xi} Q_n(\xi), \quad P_0 = 1, \tag{11.3}$$

$$Q_{n+1}(\xi) = P_n(\xi) - e^{i2\pi 2^n \xi} Q_n(\xi), \quad Q_0 = 1, \tag{11.4}$$

for $\xi \in [0; 1)$. The coefficients of the first few polynomials are

$$\begin{array}{rrrrrrrrr}
P_0: & 1 & & & & & & & \\
Q_0: & 1 & & & & & & & \\
P_1: & 1 & 1 & & & & & & \\
Q_1: & 1 & -1 & & & & & & \\
P_2: & 1 & 1 & 1 & -1 & & & & \\
Q_2: & 1 & 1 & -1 & 1 & & & & \\
P_3: & 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 \\
Q_3: & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1
\end{array} \tag{11.5}$$

It is obvious that the sequences are generated by a simple 'append' rule. We will refer to the coefficients of the RS polynomials as RS sequences. The ingenuity of these polynomials is the combination of fixed sized coefficients and the alternating sign in the recursive construction of P and Q. The former property gives

$$\|P_n\|_2^2 = \sum_{k=0}^{2^n-1} (\pm 1)^2 = 2^n, \tag{11.6}$$

while the latter property gives

$$|P_{n+1}(\xi)|^2 + |Q_{n+1}(\xi)|^2 = 2|P_n(\xi)|^2 + 2|Q_n(\xi)|^2 = 2^{n+2}, \tag{11.7}$$

since $|e^{i2\pi 2^n \xi}| = 1$. This leads to

$$|P_n(\xi)| \leq \sqrt{2} \cdot 2^{n/2}, \quad \forall \xi \in [0; 1) \ ,$$

a uniform upper bound for P_n. Now, combining (11.6) and (11.7) yields the squared crest factor

$$\frac{\|P_n\|_\infty^2}{\|P_n\|_2^2} \leq 2 \ . \tag{11.8}$$

This means that $|P_n(\xi)|^2$, $\xi \in [0; 1)$, is a function that lies within the rectangle $[0; 1] \times [0; 2^{n+1}]$, and at the same time 'covers' exactly half of its area. This guarantees the polynomial to be somewhat flat. Two examples of $|P_n|$ are shown in Fig. 11.2. At this point it is important to realize that the term 'flat' used throughout this chapter should be understood as 'not excessively far from a constant function', but not necessarily 'close to a constant function'. This was also hinted in Definition 11.4. To demonstrate the importance of this concept the two lower most graphs in Fig. 11.2 show that neither the well-known (an often used in applications) square wave nor a random ± 1 sequence can be considered flat.

11.2.2 Properties of Rudin-Shapiro Polynomials

The construction of the RS polynomial is such that the parallelogram law

$$|a + b|^2 + |a - b|^2 = 2|a|^2 + 2|b|^2$$

is the only means needed for achieving the $\sqrt{2}$ crest factor. This property is in fact essential for the relation between RS sequence and Golay complementary series. In terms of RS polynomials the law gives (11.7), i.e. that

$$P_n(\xi)\overline{P_n(\xi)} + Q_n(\xi)\overline{Q_n(\xi)} = 2^{n+1} \ .$$

Applying the inverse Fourier transform yields

$$(\mathbf{p} * \overline{\mathbf{p}})[k] + (\mathbf{q} * \overline{\mathbf{q}})[k] = 2^{n+1}\delta[k], \quad k = -2^n + 1, \ldots, 2^n - 1 \, . \tag{11.9}$$

where $\mathbf{p}$ and $\mathbf{q}$ are the coefficients sequences of P and Q, respectively, and $\overline{\mathbf{p}}$ means the time reversed of $\mathbf{p}$. Notice that (11.9) is exactly the definition of a set of complementary series.

While the crest factor of $\sqrt{2}$ was easily derived the computations leading to that result did not show whether in fact a lower bound is possible. The following lemma demonstrates that for at least some RS polynomials the crest factor is correct, i.e. the upper bound on the peak-to-mean ratio cannot be smaller.

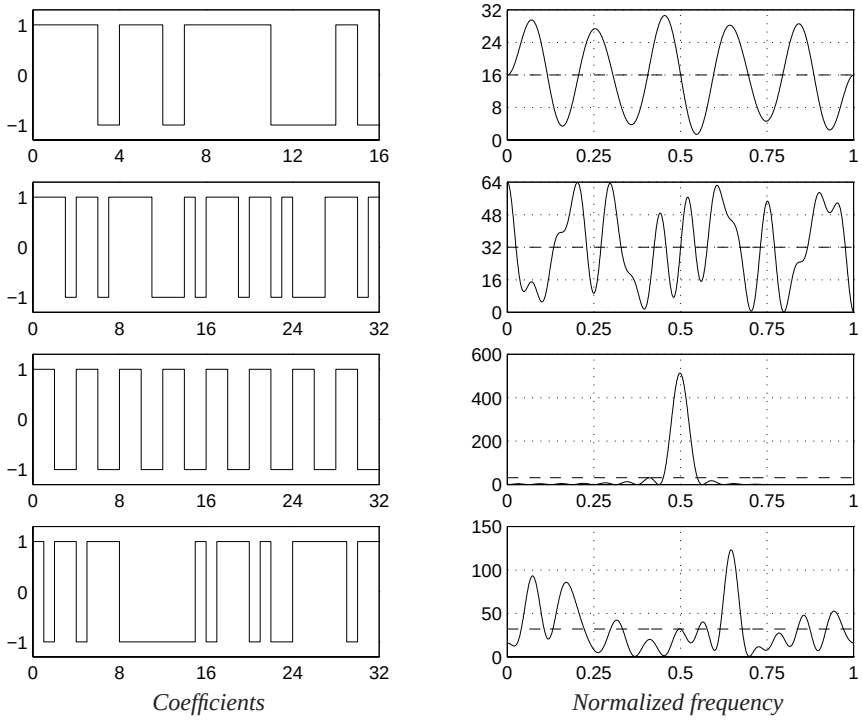


Figure 11.2: The coefficients (left) and modulo squared (right) of the Rudin-Shapiro polynomials P_4 and P_5. Below the coefficients and modulo of the Fourier transform of a square wave and a random sequence. The horizontal dashed line is the energy of the signal.

Lemma 11.6
Let P and Q be defined by (11.3) *and* (11.4)*. Then*

$$P_{2m}(0) = 2^m, \quad P_{2m}(1/2) = 2^m, \quad P_{2m+1}(0) = 2^{m+1}, \quad P_{2m+1}(1/2) = 0$$
$$Q_{2m}(0) = 2^m, \quad Q_{2m}(1/2) = -2^m, \quad Q_{2m+1}(0) = 0, \quad Q_{2m+1}(1/2) = 2^{m+1}.$$

Proof
First note that

$$\begin{aligned} P_{n+2}(\xi) &= P_{n+1}(\xi) + e^{i2\pi 2^{n+1}\xi} Q_{n+1}(\xi) \\ &= P_n(\xi) + e^{i2\pi 2^n \xi} Q_n(\xi) + e^{i2\pi 2^{n+1}\xi}\big(P_n(\xi) - e^{i2\pi 2^n \xi} Q_n(\xi)\big) \\ &= (1 + e^{i2\pi 2^{n+1}\xi}) P_n(\xi) + e^{i2\pi 2^n \xi}(1 - e^{i2^{n+1}\xi}) Q_n(\xi). \end{aligned} \tag{11.10}$$

Then for $n = 2m-2$ we have

$$\begin{aligned} P_{2m}(0) &= (1+1)P_{2m-2}(0) + 0 = \cdots = 2^m P_0(0) = 2^m, \\ P_{2m}(1/2) &= 2P_{2m-2}(1/2) = \cdots = 2^m P_0(1/2) = 2^m, \end{aligned}$$

and for $n = 2m-1$

$$\begin{aligned} P_{2m+1}(0) &= 2P_{2m-1}(0) = \cdots = 2^m P_1(0) = 2^{m+1}, \\ P_{2m+1}(1/2) &= 2P_{2m-1}(1/2) = \cdots = 2^m P_1(1/2) = 0. \end{aligned}$$

Equivalent calculations yields the results for the Q polynomials. □

The idea to these calculation is from Brillhart [8]. The P and Q polynomials are anti-symmetric around $1/4$.

Lemma 11.7
Let $\mathbf{p}$, $\mathbf{q}$ *be two Rudin-Shapiro sequences. Then*

$$\begin{aligned} |P_n(\xi)|^2 &= 2^{n+1} - |P_n(1/2 - \xi)|^2 \\ |Q_n(\xi)|^2 &= 2^{n+1} - |Q_n(1/2 - \xi)|^2. \end{aligned}$$

Proof
The lemma obviously holds for $n = 0$. Then the result follows from an induction argument.

$$\begin{aligned} |P_{n+1}(\xi)|^2 &= |P_n(\xi)|^2 + |Q_n(\xi)|^2 + e^{i2\pi 2^n \xi}\overline{P_n(\xi)}Q_n(\xi) + e^{-i2\pi 2^n \xi} P_n(\xi)\overline{Q_n(\xi)} \\ &= 2^{n+1} - |P_n(1/2-\xi)|^2 + 2^{n+1} - |Q_n(1/2-\xi)|^2 \\ &\quad + 2\,\mathrm{Re}\big\{e^{i2\pi 2^n \xi}\overline{P_n(\xi)}Q_n(\xi)\big\} \\ &= 2^{n+2} - |P_n(1/2-\xi)|^2 - |Q_n(1/2-\xi)|^2 \end{aligned}$$

$$
\begin{aligned}
&\quad - 2\,\mathrm{Re}\left\{e^{i2\pi 2^n(1/2-\xi)}\overline{P_n(1/2-\xi)}\,Q_n(1/2-\xi)\right\} \\
&= 2^{n+2} - |P_{n+1}(1/2-\xi)|^2 .
\end{aligned}
$$

Since P and Q are trigonometric polynomials the third equality is given by a calculation that involves the cosine equality $\cos(\xi) = -\cos(\pi - t)$. □

The following lemma shows that the append rule presented for the Rudin-Shapiro sequences which is used to produce longer sequences, actually apply to all complementary sequences.

Lemma 11.8
Let $\mathbf{p}, \mathbf{q} \in \mathbb{C}^n$ be two vectors with the properties

$$\langle \tau_{2k}\mathbf{p}, \mathbf{q}\rangle = 0, \qquad \langle \tau_{2k}\mathbf{p}, \mathbf{p}\rangle = \langle \tau_{2k}\mathbf{q}, \mathbf{q}\rangle = C\delta[k],$$

where τ_m means a shift of index by $+m$. Define

$$\tilde{\mathbf{p}} = \begin{bmatrix}\mathbf{p}\\ \mathbf{q}\end{bmatrix} \quad \textit{and} \quad \tilde{\mathbf{q}} = \begin{bmatrix}\mathbf{p}\\ -\mathbf{q}\end{bmatrix}.$$

Then

$$\left\langle \tau_{2k}\tilde{\mathbf{p}}, \tilde{\mathbf{q}}\right\rangle = 0, \qquad \left\langle \tau_{2k}\tilde{\mathbf{p}}, \tilde{\mathbf{p}}\right\rangle = \left\langle \tau_{2k}\tilde{\mathbf{q}}, \tilde{\mathbf{q}}\right\rangle = 2C\delta[k].$$

Note that $\langle \tau_{2k}\mathbf{p}, \mathbf{q}\rangle = (\mathbf{p} * \bar{\mathbf{p}}[-2k]$.

Proof
From the definitions of $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{q}}$ it follows that

$$
\left.\begin{matrix}\left\langle \tau_{2k}\tilde{\mathbf{p}}, \tilde{\mathbf{q}}\right\rangle \\ \left\langle \tau_{2k}\tilde{\mathbf{p}}, \tilde{\mathbf{p}}\right\rangle \\ \left\langle \tau_{2k}\tilde{\mathbf{q}}, \tilde{\mathbf{q}}\right\rangle\end{matrix}\right\} =
\begin{cases}
\pm\langle \tau_{2k+N}\mathbf{p}, \mathbf{q}\rangle & \text{for } k = -N+1, \ldots, -N/2, \\
\langle \tau_{2k}\mathbf{p}, \mathbf{p}\rangle \pm \langle \tau_{2k}\mathbf{q}, \mathbf{q}\rangle \pm \langle \tau_{2k+N}\mathbf{p}, \mathbf{q}\rangle & \text{for } k = -N/2+1, \ldots, -1, \\
\langle \tau_{2k}\mathbf{p}, \mathbf{p}\rangle \pm \langle \tau_{2k}\mathbf{q}, \mathbf{q}\rangle \pm \langle \tau_{2k-N}\mathbf{p}, \mathbf{q}\rangle & \text{for } k = 1, \ldots, N/2-1, \\
\pm\langle \tau_{2k-N}\mathbf{p}, \mathbf{q}\rangle & \text{for } k = N/2, \ldots, N-1.
\end{cases}
$$

All four expressions equal zero independently of the signs. For the zero shift

$$\left\langle \tilde{\mathbf{p}}, \tilde{\mathbf{q}}\right\rangle = \langle \mathbf{p}, \mathbf{p}\rangle - \langle \mathbf{q}, \mathbf{q}\rangle = 0\ ,$$

and

$$\left\langle \tilde{\mathbf{p}}, \tilde{\mathbf{p}}\right\rangle = \left\langle \tilde{\mathbf{q}}, \tilde{\mathbf{q}}\right\rangle = \langle \mathbf{p}, \mathbf{p}\rangle + \langle \mathbf{q}, \mathbf{q}\rangle = 2C\ .$$

□

An obvious consequence of this lemma is

Corollary 11.8.1
Any Rudin-Shapiro sequence set $\mathbf{p}, \mathbf{q}$ have the property $\langle \tau_{2k}\mathbf{p}, \mathbf{q}\rangle = 0$.

A more general statement about the autocorrelation of RS sequences is given in Taghavi [76] and [75]. The results are presented in the following lemma.

Lemma 11.9
Let $\mathbf{p}$ *be a RS sequence of length* 2^N*. Then*

$$\left|\langle \tau_k \mathbf{p}, \mathbf{p}\rangle\right| \leq 3.2134 \cdot 2^{0.7303N}$$

for $k = -N+1, \ldots, N-1$*. Further there exists* C *such that*

$$\left|\langle \tau_k \mathbf{p}, \mathbf{p}\rangle\right| > C 2^{0.73N} \ .$$

Finally, the following theorem provides a lower bound to the ratio between the cardinality of $\{\mathbf{c} \in \mathscr{S}_N^2 | \mathcal{C}(\mathbf{c}) \leq \sqrt{2}\}$ and $\mathscr{S}_N^2$ itself.

Theorem 11.10
There are at least $(2 - \delta_K + o(1))^n$ *functions* $f_n \in \mathscr{H}_n^2$ *with* $\|f\|_\infty \leq K\|f\|_2$*. The* δ_k *is defined for all* $K \geq K_0$*,* K_0 *being an absolute constant. For these* K*,* $0 < \delta_K < 1$*. Furthermore* $\lim_{K\to\infty} \delta_K = 0$*. Here* $o(1)$ *is a function approaching* 0 *as* $n \to \infty$ *for fixed* K*.*

This theorem is due to Spencer [72], who presents it as a corollary to a theorem on a two-coloring problem. He also conjectures that the number of Rudin-Shapiro like functions are bounded from above by $(2 - \varphi_K + o(1))^n$, for some $\varphi_K > 0$.

11.3 The Rudin-Shapiro Transform

An interesting property of the RS sequences generated according to the appending rule in (11.3) and (11.4) is that they are orthogonal. This is immediately evident from the appending example shown. It is also worth noting that interchanging the $+$ and $-$ in (11.3) and (11.4) would still produce sequences with all the previously presented properties. In fact, arbitrarily interchanges of the signs in each recursive step does not affect the properties of the constructed sequences.

An elegant construction achieving all combinations of sign changes is found in Benke [6] (Byrnes [11, 13] gives a similar construction). In short,

$$\begin{bmatrix} P_{n+1,\epsilon}(\xi) \\ Q_{n+1,\epsilon}(\xi) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^{\epsilon_n} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{i2\pi 2^n \xi} \end{bmatrix} \begin{bmatrix} P_{n,\epsilon}(\xi) \\ Q_{n,\epsilon}(\xi) \end{bmatrix}, \tag{11.11}$$

where $\epsilon_n \in \{0, 1\}$ is chosen in each step. A total of 2^n different P polynomials are possible after n steps. Thus, two P polynomials with each two coefficients are obtained after one steps, four P polynomials with each four coefficients are obtained after two steps, and so on. The two and four polynomials have coefficients

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 \end{bmatrix},$$

and the eight P polynomials after the third step have coefficients

$$\begin{bmatrix} 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 \\ 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & 1 & -1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 \end{bmatrix}$$

Note that all rows in the matrices are orthogonal. Thus, the RS sequences of length 2^J constitutes an orthogonal basis of $\mathbb{R}^{2^J}$. Consequently, the matrices are called the Rudin-Shapiro transform (RST). It is shown in Benke [6] that this construction can be generalized in various ways.

An interesting property derived by the author is the following. The individual entries in the Rudin-Shapiro transform can be found by the following equation, where $\mathbf{P}^{(N)} \equiv [p_{m,n}^{(N)}]$ is the $2^N \times 2^N$ RST matrix.

$$p_{m,n}^{(N)} = \prod_{k=1}^{N} (-1)^{n_k(m_{N-k+1}+n_{k-1})}, \quad n_0 \equiv 0.$$

where n_k and m_k is the k'th binary digit of n and m respectively, with $k = 1$ as LSB. This property is not proved at this points as a very similar equation is given and proved in the next section.

Applying the RST decomposes a signal into a basis of elements with a spread spectrum property. This is in some sense the opposite of a Fourier transform which is a decomposition into a narrow spectrum basis. The transform is orthogonal and thus energy preserving, and the equal amplitude of all the entries makes the transform numerical stable. In general, it is an appealing transform for design and analysis of spread spectrum signals. However, at this point a fast implementation is still missing. Matrix multiplication is a $O(N^2)$ operation, and in general it is preferable, if not desirable, to have an $O(N \log N)$ implementation, especially for real time applications.

Note also that while the rows of the presented matrices do have a low crest factor, this is not the case for the columns which exhibits a Walsh-like structure rather than spread spectrum structure.

The problems mentioned here are addressed in the following section, where a slight change of the recursive definition of the RS polynomials yields a symmetric RS transform. At the same time a fast implementation, actually $O(N \log N)$ with a small constant, is also given.

11.4 The Symmetric Rudin-Shapiro Transform

The Rudin-Shapiro transform can be made symmetric. The idea for this is communicated in Byrnes et al. [13]. There the polynomials are defined by a modification of the previously presented definition in (11.3) and (11.4). The following equations have been slightly rewritten compared to [13], to comply with the notation in this chapter (most significantly, Byrnes have discarded the Q polynomials in favor of a more advanced indexing of the P polynomials). The symmetric RST is derived from the following equations.

$$\begin{aligned}
P_{j+1,4m}(\xi) &= P_{j,2m}(\xi) + e^{i2\pi 2^j\xi} Q_{j,2m+1}(\xi),\\
P_{j+1,4m+1}(\xi) &= P_{j,2m}(\xi) - e^{i2\pi 2^j\xi} Q_{j,2m}(\xi),\\
P_{j+1,4m+2}(\xi) &= P_{j,2m+1}(\xi) + e^{i2\pi 2^j\xi} Q_{j,2m+1}(\xi),\\
P_{j+1,4m+3}(\xi) &= -P_{j,2m+1}(\xi) + e^{i2\pi 2^j\xi} Q_{j,2m+1}(\xi),\\
Q_{j+1,4m}(\xi) &= P_{j,2m}(\xi) - e^{i2\pi 2^j\xi} Q_{j,2m}(\xi),\\
Q_{j+1,4m+1}(\xi) &= P_{j,2m}(\xi) + e^{i2\pi 2^j\xi} Q_{j,2m}(\xi),\\
Q_{j+1,4m+2}(\xi) &= -P_{j,2m+1}(\xi) + e^{i2\pi 2^j\xi} Q_{j,2m+1}(\xi),\\
Q_{j+1,4m+3}(\xi) &= P_{j,2m+1}(\xi) + e^{i2\pi 2^j\xi} Q_{j,2m+1}(\xi),
\end{aligned} \tag{11.12}$$

with

$$P_{1,0} = Q_{1,1} = 1 + e^{i2\pi\xi} \quad \text{and} \quad P_{1,1} = Q_{1,0} = 1 - e^{i2\pi\xi},$$

and for $j \geq 1$ and $m = 0, \ldots, 2^{j-1} - 1$. Note that P and Q in (11.12) are equal to the previous definition in (11.3) and (11.4) except for some changes of signs. The properties derived in the previous sections therefore still applies.

It is not proven in [13] that this definition leads to a symmetric transform. Neither does it contain a clear description of how to apply the transform to a signal. This section is therefore dedicated to a rigorous proof of the symmetry (and the other desirable properties of the symmetric RST). The proof is 'constructive' in that it provides a simple way of applying the transform, namely by means of the Haar wavelet packet transform scheme.

11.4.1 Deriving the Symmetric Transform

The equations (11.12) can be written more compactly as

$$P_{j+1,m}(\xi) = (-1)^{m_1 m_2} P_{j,\lfloor m/2\rfloor}(\xi) + (-1)^{m_1(m_2+1)} e^{i2\pi 2^j\xi} Q_{j,\lfloor m/2\rfloor}(\xi), \tag{11.13}$$

$$Q_{j+1,m}(\xi) = (-1)^{(m_1+1)m_2} P_{j,\lfloor m/2\rfloor}(\xi) + (-1)^{(m_1+1)(m_2+1)} e^{i2\pi 2^j\xi} Q_{j,\lfloor m/2\rfloor}(\xi), \tag{11.14}$$

where m_1 and m_2 are the two least significant digits of the binary representation of m, and $\lfloor m/2\rfloor$ means the biggest integer less or equal to $m/2$. Rewriting to the obvious matrix

form yields

$$\begin{bmatrix} P_{j+1,m}(\xi) \\ Q_{j+1,m}(\xi) \end{bmatrix} = \begin{bmatrix} (-1)^{m_1 m_2} & (-1)^{m_1(m_2+1)} \\ (-1)^{(m_1+1)m_2} & (-1)^{(m_1+1)(m_2+1)} \end{bmatrix} \begin{bmatrix} P_{j,\lfloor m/2 \rfloor}(\xi) \\ e^{i2\pi 2^j \xi} Q_{j,\lfloor m/2 \rfloor}(\xi) \end{bmatrix}. \quad (11.15)$$

This latter form of the RS equations shows the core of the transform; the 2×2 matrix. Incidentally, this is also the 'secret' of the easy implementation.

To have a solid basis for the derivation of the RST properties, the first thing to do is define exactly what the RST is.

Definition 11.11 (The Symmetric Rudin-Shapiro Transform)
Define the mapping $\mathbf{P}_{j,m} : \mathbb{R}^{2^j} \mapsto \mathbb{R}^{2^j}$, $j \geq 1$, *as*

$$\begin{bmatrix} y_k \\ y_{k+2^{j-1}} \end{bmatrix} = \frac{(-1)^{mk}}{\sqrt{2}} \begin{bmatrix} 1 & (-1)^k \\ (-1)^m & -(-1)^{k+m} \end{bmatrix} \begin{bmatrix} x_{2k} \\ x_{2k+1} \end{bmatrix} \quad (11.16)$$

for $k = 0, \ldots, 2^{j-1} - 1$ *when mapping* $\mathbf{x}$ *to* $\mathbf{y}$. *Define*

$$\mathbf{P}_j^{(J)} \equiv \begin{bmatrix} \mathbf{P}_{j,0} & & 0 \\ & \ddots & \\ 0 & & \mathbf{P}_{j,2^{J-j}-1} \end{bmatrix}, \quad (11.17)$$

and finally defined the Rudin-Shapiro transform (RST) $\mathbf{P}^{(J)}$ *and the auxiliary transform* $\mathbf{Q}^{(J)}$ *as*

$$\mathbf{P}^{(J)} \equiv \prod_{j=1}^{J} \mathbf{P}_j^{(J)}, \qquad \textit{and} \qquad \mathbf{Q}^{(J)} \equiv \prod_{j=1}^{J-1} \mathbf{P}_j^{(J)} \mathbf{P}_{J,1}. \quad (11.18)$$

Note that (11.16) is the inverse of the transform proposed in (11.15). The 2×2 matrix in (11.16) aside, it is not immediately obvious neither how this definition is linked to (11.12), nor that it defines a symmetric transform. However, the following theorem establishes that this definition does indeed provide the desired transform.

Theorem 11.12 (Properties of the Rudin-Shapiro Transform)
The Rudin-Shapiro transform $\mathbf{P}^{(J)} : \mathbb{R}^{2^J} \mapsto \mathbb{R}^{2^J}$ *and the corresponding polynomials*

$$P_m^{(J)}(\xi) = \sum_{n=0}^{2^J-1} p_{m,n}^{(J)} e^{i2\pi n\xi}.$$

has the following properties:

(I) The rows of $\mathbf{P}^{(J)}$ *are the coefficients of the polynomials defined in* (11.12).

(II) *The entries of* $\mathbf{P}^{(J)} = \left[p_{m,n}^{(J)}\right]$ *are given by*

$$p_{m,n}^{(J)} = 2^{-J/2} \prod_{j=1}^{J} (-1)^{(m_j + n_{J-j+2})(m_{j+1} + n_{J-j+1})}, \qquad (11.19)$$

for $m, n = 0, \ldots, 2^J - 1$, *where* m_j *are the* j*'th digit in the binary representation of* m, *with* m_1 *LSB.*

(III) *It is a unitary and symmetric Hadamard matrix.*

(IV) *It satisfies*[*]

$$0 < |P_m^{(J)}(\xi)| < \sqrt{2}, \quad m = 0, \ldots, 2^J - 1, \qquad (11.20)$$

on $(0; 1/2)$. *Moreover,*

$$P_{2j}(0) = P_{2j}(1/2) = 1, \qquad (11.21)$$

and

$$P_{2j+1}(0) = \sqrt{2}, \quad P_{2j+1}(1/2) = 0, \qquad (11.22)$$

and finally

$$P_j(1/4) = 1\,. \qquad (11.23)$$

Proof
To prove (I) first note

$$\begin{aligned} \mathbf{P}^{(j)} &= \left(\mathbf{P}_{j,0}\right)^{\top} \begin{bmatrix} \mathbf{P}^{(j-1)} & \\ & \mathbf{Q}^{(j-1)} \end{bmatrix}, \\ \mathbf{Q}^{(j)} &= \left(\mathbf{P}_{j,1}\right)^{\top} \begin{bmatrix} \mathbf{P}^{(j-1)} & \\ & \mathbf{Q}^{(j-1)} \end{bmatrix}. \end{aligned} \qquad (11.24)$$

This follows from

$$\begin{aligned} &\left(\mathbf{P}_{j,0}\right)^{\top} \begin{bmatrix} \mathbf{P}^{(j-1)} & \\ & \mathbf{Q}^{(j-1)} \end{bmatrix} \\ &\quad = \left(\mathbf{P}_{j,0}\right)^{\top} \begin{bmatrix} \mathbf{P}_{j-1,0} & \\ & \mathbf{P}_{j-1,1} \end{bmatrix}^{\top} \begin{bmatrix} \mathbf{P}^{(j-2)} & & & \\ & \mathbf{Q}^{(j-2)} & & \\ & & \mathbf{P}^{(J-2)} & \\ & & & \mathbf{Q}^{(j-2)} \end{bmatrix} \\ &\qquad \vdots \end{aligned}$$

[*]Only semi-flatness, and not near-flatness of the polynomials is actual proven in this thesis. However, the author feels sufficiently confident about the validity of the statement to include it in the theorem.

$$= \prod_{k=j}^{1} (\mathbf{P}_k^{(j)})^\top$$
$$= \mathbf{P}^{(j)}.$$

Note also that $\left(\mathbf{P}_{j,0}\right)^\top$ is the transform given as

$$\begin{bmatrix} x_{2k} \\ x_{2k+1} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ (-1)^k & -(-1)^k \end{bmatrix} \begin{bmatrix} y_k \\ y_{k+2^{j-1}} \end{bmatrix}$$

for $k = 0, \ldots, 2^{j-1} - 1$ when mapping $\mathbf{y}$ to $\mathbf{x}$. So

$$\left(\mathbf{P}_{j,0}\right)^\top = \frac{1}{\sqrt{2}} \begin{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} & & & \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} & & \\ & \ddots & & & \ddots & \\ & & \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} & & & \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \end{bmatrix}_{2^j \times 2^j}$$

Letting $\mathbf{p}_m^{(j)}$ denote the m'th row of $\mathbf{P}^{(j)}$, and likewise with $\mathbf{Q}^{(j)}$, it follows that

$$\mathbf{P}^{(j)} = \left(\mathbf{P}_{j,0}\right)^\top \begin{bmatrix} \mathbf{P}^{(j-1)} & \\ & \mathbf{Q}^{(j-1)} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{p}_0^{(j)} & \mathbf{q}_0^{(j)} \\ \mathbf{p}_0^{(j)} & -\mathbf{q}_0^{(j)} \\ \mathbf{p}_1^{(j)} & \mathbf{q}_1^{(j)} \\ -\mathbf{p}_1^{(j)} & \mathbf{q}_1^{(j)} \\ \vdots & \vdots \\ \mathbf{p}_{2^j-2}^{(j)} & \mathbf{q}_{2^j-2}^{(j)} \\ \mathbf{p}_{2^j-2}^{(j)} & -\mathbf{q}_{2^j-2}^{(j)} \\ \mathbf{p}_{2^j-1}^{(j)} & \mathbf{q}_{2^j-1}^{(j)} \\ -\mathbf{p}_{2^j-1}^{(j)} & \mathbf{q}_{2^j-1}^{(j)} \end{bmatrix}, \tag{11.25}$$

which demonstrates the appending rule defined in the first four equations of (11.12). A similar calculation will show the last four equations.

The proof of (II) goes by induction on (11.19). In the following the scaling $2^{-J/2}$ is ignored. For $J = 1$

$$\mathbf{P}^{(1)} = \begin{bmatrix} p_{0,0}^{(1)} & p_{0,1}^{(1)} \\ p_{1,0}^{(1)} & p_{1,1}^{(1)} \end{bmatrix} = \begin{bmatrix} (-1)^{(0+0)(0+0)} & (-1)^{(0+0)(0+1)} \\ (-1)^{(1+0)(0+0)} & (-1)^{(1+0)(0+1)} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

which is correct according to (11.16). Assume that (11.19) is true for j. From (11.25) it is seen that

$$p_{m,n}^{(j+1)} = \begin{cases} (-1)^{m_2 m_1} p_{\lfloor m/2 \rfloor, n}^{(j)} & n = 0, \ldots, 2^j - 1 \\ (-1)^{(m_2+1)m_1} q_{\lfloor m/2 \rfloor, n-2^j}^{(j)} & n = 2^j, \ldots, 2^{j+1} - 1. \end{cases} \tag{11.26}$$

The first case can be rewritten

$$\begin{aligned} (-1)^{m_2 m_1} p^{(j)}{}_{\lfloor m/2 \rfloor, n} &= (-1)^{m_2 m_1} \prod_{k=1}^{j} (-1)^{(m_{k+1}+n_{j-k+2})(m_{k+2}+n_{j-k+1})} \\ &= (-1)^{(m_1+n_{j+2})(m_2+n_{j+1})} \prod_{k=2}^{j+1} (-1)^{(m_k+n_{j+1-k+2})(m_{k+1}+n_{j+1-k+1})} \\ &= \prod_{k=1}^{j+1} (-1)^{(m_k+n_{j+1-k+2})(m_{k+1}+n_{j+1-k+1})}, \quad n = 0, \ldots 2^j - 1. \end{aligned}$$

To rewrite the second case, the connection between $p_{m,n}^{(j)}$ and $q_{m,n}^{(j)}$ are derived. From (11.12) it is seen that

$$\begin{aligned} Q_{j+1,4k}(\xi) &= P_{j+1,4k+1}(\xi), \\ Q_{j+1,4k+1}(\xi) &= P_{j+1,4k}(\xi), \\ Q_{j+1,4k+2}(\xi) &= P_{j+1,4k+3}(\xi), \\ Q_{j+1,4k+3}(\xi) &= P_{j+1,4k+2}(\xi) \end{aligned} \tag{11.27}$$

Changing the sign in this manner can be accomplished by adding 1 to the LSB of the row counter variable, that is to m_1. Thus,

$$q_{m,n}^{(j)} = (-1)^{(m_1+1)(m_2+n_j)} \prod_{k=2}^{j} (-1)^{(m_k+n_{j-k+2})(m_{k+1}+n_{j-k+1})},$$

and the second case of (11.26) can be rewritten

$$\begin{aligned} &(-1)^{(m_2+1)m_1} q_{\lfloor m/2 \rfloor, n-2^j}^{(j)} \\ &\quad = (-1)^{(m_2+1)m_1} (-1)^{(m_2+1)(m_3+n_j)} \prod_{k=2}^{j} (-1)^{(m_{k+1}+n_{j-k+2})(m_{k+2}+n_{j-k+1})} \\ &\quad = (-1)^{(m_1+n_{j+2})(m_2+n_{j+1})} (-1)^{(m_2+n_{j+1})(m_3+n_j)} \\ &\qquad\qquad \times \prod_{k=3}^{j+1} (-1)^{(m_k+n_{j+1-k+2})(m_{k+1}+n_{j+1-k+1})} \end{aligned}$$

$$= \prod_{k=1}^{j+1} (-1)^{(m_k + n_{j+1-k+2})(m_{k+1} + n_{j+1-k+1})}, \quad n = 2^j, \ldots 2^{j+1} - 1.$$

The second last equality is due to $n_{j+1} = 1$ and $n_{j+2} = 0$. This proves (11.19).

The unitarity of $\mathbf{P}^{(J)}$ stated in (III) follows immediately from unitarity of $\mathbf{P}_{j,m}$, and according to (II) $\mathbf{P}_{(J)}$ is a Hadamard matrix. Only the symmetry remains to be established. By interchanging m and n in the power of (-1) in (11.19), and substituting $k = J - j + 1$

$$\begin{aligned} p_{n,m}^{(J)} &= 2^{-J/2} \prod_{j=1}^{J} (-1)^{(n_j + m_{J-j+2})(n_{j+1} + m_{J-j+1})} \\ &= 2^{-J/2} \prod_{k=J}^{1} (-1)^{(n_{J-k+1} + m_{k+1})(n_{J-k+2} + m_k)} \\ &= p_{m,n}^{(J)} \end{aligned}$$

so interchanging m and n in (11.19) is equivalent to reversing the order of multiplication. It follows that the matrix $\mathbf{P}^{(N)}$ is symmetric.

The near-flat polynomial property in (IV) has already been demonstrated for as far as semi-flatness goes. However, despite a significant effort the attempt of the author to find a proof of near-flatness of the polynomials on $(0; 1/2)$ have been fruitless.

The equations (11.21) and (11.22) follows from a series of calculations equivalent to Lemma 11.6. A rewriting of (11.13) in the same fashion as (11.10) yields

$$\left.\begin{matrix} P_{j+2,m}(\xi) \\ Q_{j+2,m}(\xi) \end{matrix}\right\} = \left(\pm 1 \pm e^{i2\pi 2^{j+1}\xi}\right) P_{j,u}(\xi) + e^{i2\pi 2^j \xi} (\pm 1 \pm e^{i2\pi 2^{j+1}\xi}) Q_{j,u}(\xi)$$

where the two signs inside each of the parentheses will be the same in the one and opposite in the other parenthesis. Thus,

$$\begin{aligned} \left.\begin{matrix} P_{j+2,m}(1/4) \\ Q_{j+2,m}(1/4) \end{matrix}\right\} &= \left(\pm 1 \pm e^{i2\pi 2^{j-1}}\right) P_{j,u}(1/4) + e^{i2\pi 2^{j-2}} (\pm 1 \pm e^{i2\pi 2^{j-1}}) Q_{j,u}(1/4) \\ &= \begin{cases} \pm 2 P_{j,u}(1/4) & \text{for some } m \\ \pm 2 Q_{j,u}(1/4) & \text{for the other } m \ . \end{cases} \end{aligned}$$

Then

$$P_{2n,m}(1/4)| = |Q_{2n,m}(1/4)| = 2^{n-1}|P_{2,u}| = 2^{n-1}|Q_{2,u}| = 2^n$$

and

$$P_{2n-1,m}(1/4)| = |Q_{2n-1,m}(1/4)| = 2^{n-1}|P_{1,u}| = 2^{n-1}|Q_{1,u}| = \sqrt{2} \cdot 2^{n-1} \ .$$

This proves (11.23). □

11.4.2 Fast Implementation

The definition of the RST given in Definition 11.11 is based on the recursive construction process of RS polynomials. When writing this process in matrix form the 2×2 matrix in (11.16) emerges along with the $2^J \times 2^J$ matrix in (11.17). The combination of these two matrices is actually the key to a fast implementation. The large matrix is a factorization of the RST matrix, and the small matrix gives a simple and easy implementation of the large matrix.

The factorization means that the RST can be applied in J steps by multiplying a signal with all of the $\mathbf{P}_j^{(J)}$ matrices (in the right order). Each multiplication is an $O(N^2)$ operation, but the mapping given in (11.16) shows how to reduce the multiplication to an $O(N)$ filtering process. For any choice of m and k the 2×2 matrix contains 3 times $+1$ and one -1. Consequently, the output of the mapping is merely a series of sums and differences of sample pairs. A division by $\sqrt{2}$ should be applied to every sum/difference, but since the mapping is linear this scaling can be applied as division by 2 for every other step in the transform. Note that division by 2 is equivalent to a binary shift of 1.

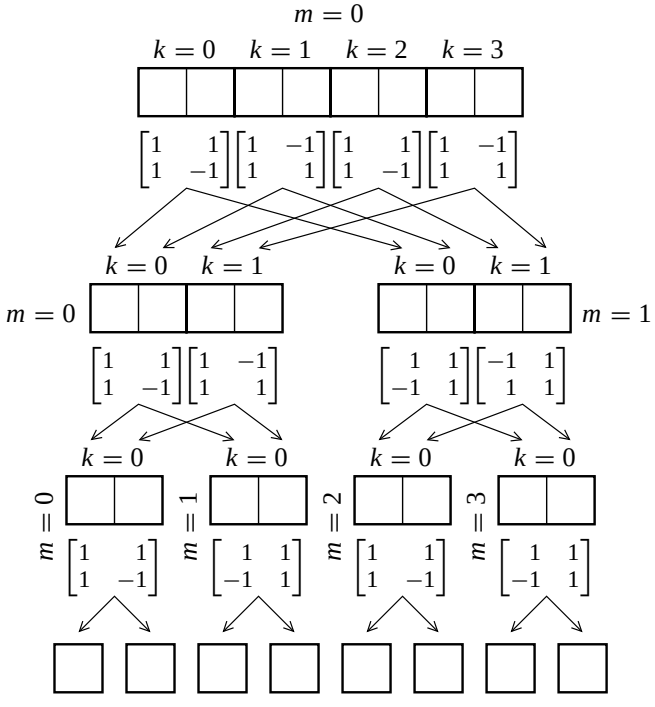


Figure 11.3: This figure shows how the value of the variables change in the fast implementation of a symmetric RST. Here applied to a vector in $\mathbb{R}^8$.

When implementing the RST according to this scheme it is obviously important to get the 2×2 matrix correct. The m and k change constantly as the transform is applied. In Fig. 11.3 these changes are shown along with the 2×2 matrix for each sample pair in each step of the transform.

It is not apparent from this visualization of the fast implementation that it is its own inverse, i.e. if the resulting signal at the bottom is placed at the top, the new output is actually the original signal. But as it as been shown previously this is indeed the case since the transform is symmetric.

Suppose that the same matrix is used throughout, i.e. suppose that m and k equal zero in all cases. The result is the a full decomposition wavelet packet Haar transform. The Haar transform is also its own inverse. If only m equals zero the result is the non-symmetric RST presented in (11.11). This is easily seen as

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^{\epsilon_n} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & (-1)^{\epsilon_n} \\ 1 & -(-1)^{\epsilon_n} \end{bmatrix}.$$

The relation to the Haar transform can also provide an explanation for the spread spectrum property without involving the RS polynomials. The Haar transform is a decomposition into a frequency localizing basis since the Haar filters are low and high pass filters (with two filter taps). This means that each element in the output from the (full decomposition) Haar transform represents the energy in a certain frequency range of the original signal. The RST does in some sense the exact opposite of the this. Instead of applying the same filter to all samples pairs (and thereby creating a output localized in frequency) the RST applies the low and high pass filters alternately to sample pairs. The result is an output where the samples are the same as in the Haar transform case, but where they are mixed such that there are no frequency localization at all.

11.4.3 Other Properties of Rudin-Shapiro Polynomials

The work with RS polynomials and sequence have led the author to believe in some other properties for which no proof have yet been devised. These results are presented here as conjectures, and without any further explanations. So far, these results have found no practical use.

The first conjecture states that although the individual RS polynomials are (supposedly) near-flat on $(0;\, 1/2)$ they are not flat on $(0;\, 1/2)$.

Conjecture 11.13
Let $P_{j,m}(\xi)$ be one of the polynomials defined in (11.12)*. Then*

$$\lim_{j\to\infty} 2^{-j/2} \max_{\xi\in(0;1/2)} |P_{j,m}(\xi)| = \sqrt{2}$$

and the convergence is of order $O(e^{-j})$.

It seems that polynomials are equal in equidistant points with a finer resolution for longer polynomials

Conjecture 11.14
Let $P_{j,k}(\xi)$ be one of the polynomials defined in (11.12)*. Then*

$$2|P_{j,k}\big(m2^{-j}\big)| = |P_{j+2,k}\big(m2^{-j}\big)|, \quad k, m = 0, \ldots, 2^j - 1 \, .$$

In the limit this 'result' becomes

Conjecture 11.15
The limits

$$\lim_{j\to\infty} 2^{-j} P_{2j,k}(\xi) \qquad \textit{and} \qquad \lim_{j\to\infty} 2^{-j} P_{2j+1,k}(\xi)$$

converge pointwise on the dense subset $\big\{m2^{-n};\, m = 0, \ldots, 2^n\big\}_{n\in\mathbb{N}}$ of the unit interval.

The recursive construction of the polynomials means that there are many different relations between the various polynomials. A few has been conjectured upon here, and others can easily be discovered by experiments. The next chapter is dedicated to an investigation of the self-similarity of the RST which is inherited from the original definition of the RS polynomials.

Linear Transform of the Rudin-Shapiro Matrix

12

With an invertible transform a signal can be decomposed into a set of coefficients, and subsequently reconstructed completely using those same coefficients. However, if the coefficients are subject to some alteration, not only is this property in general lost, but the 'reconstruction' might produce a completely different signal. If the nature of the alteration is a priori known it is possible to predict the impact of the alteration, and in some cases the prediction is quite easy to make, and the alteration can perhaps be undone.

The RST behaves in a simple and predictable way for alterations which are made when a block diagonal linear transform is applied to the transformed signal. Constructing and understanding the structure of this prediction not as easy, though.

The first section presents an example which illustrates the principle of and motivation for the subject of this chapter. Although the point of the example has to some extent already been delivered by the test signals in the third setup, see Section 5.4, and although a brief discussion of the usefulness of the main result was given in Section 4.8.2, the author feels that further motivation would not come amiss. Especially, since this example is focuses on the subject of this chapter.

The second and final section of this chapter shows how the self-similarity properties of the RST are used to predict the impact of applying block diagonal linear transforms to RS transformed signals.

12.1 Motivation

Consider a discrete signal consisting of 512 samples with all but sample 351 vanishing. The RST of this signal is shown in Fig. 12.1(a), and the power spectrum is shown in (b). This clearly demonstrates that the RS sequence in (a) is coefficients of a flat polynomial. The RS sequence is now used for modulating an LED according to the scheme described in Section 4.1.1. Some distance away a photo diode is located which converts the light into an electric signal. In Chapter 5 a number of detailed examples are given on test setups for recording this type of signals. The received signal is shown in Fig. 12.1(c). The task is now to determine the CGM, i.e. the intensity of the received RS sequence. Since this signal is recorded in a room with artificial lighting (which is much more intensive that the LED), most of the energy does not come from the LED. Actually, the RS sequence

is not visible with the current scaling of the Y-axis in (c). Even the tiny ripples is not the RS sequence, but different types of noise. Consequently, computing the (inverse) RST reveals no trace of the RS sequence, as shown in (d): Sample 351 should have been clearly distinguishable.

The major disturbance in (c) is the sinusoid-like shape of the signal, which originates in the 100 Hz artificial lighting. There are several ways of removing this disturbance from the recorded signal. One possibility is to consider (c) as a sinusoid, track the phase and amplitude, and subsequently remove that particular sine component. This approach was discussed in Section 4.7.1. However, this will reveal, see (e), that the dominant structure of (c) is not a pure sinusoid. This is due to physical and electrical phenomenon in the receiving electronic circuit.

Two other approaches are a high pass filtering and removal of low-degree polynomial content. Both operations are linear, and it would be nice to be able to predict what effect such operations have on the RS sequence, which is hidden in (c). In (f) the signal in (c) has been denoised by dividing it into 16 consecutive parts of 32 samples. Each part has been projected onto a space spanned by sampled Legendre polynomials of degree 4 through 15 (thus removing all polynomial content of degree 0 to 3). The 16 resulting pieces have then been concatenated to create the signal in (f). An RST of this signal is shown in (g), and sample 351 is now clearly visible. If one further wants to know the amplitude of the RS sequence and an estimated uncertainty of the amplitude, it is vital to realize what the previously applied linear transform, call it $\mathbf{L}$, does to the RS sequence.

In this case $\mathbf{L}$ has altered the RS sequence $\mathbf{x}$ such that $\mathbf{P}^{(9)}\mathbf{L}\mathbf{x}$ has exactly 64 non-vanishing entries (and in general there are 2^{J-s+1} non-vanishing entries, where 2^J is the size of the RST and 2^s is the number of signal parts), where $\mathbf{P}^{(9)}\mathbf{x}$ has just one, namely entry 351. An interesting point to notice is that there are 448 entries which are unaffected by the denoising. The position of the 64 affected entries are shown with small dots in Fig. 12.1(g). The unaffected samples provide information on the uncertainty of the actual value of entry 351. Methods for using this information for validating the CGM was discussed in Section 4.9. Moreover,

$$\frac{\mathbf{P}^{(9)}\mathbf{L}\mathbf{x}[351]}{\mathbf{P}^{(9)}\mathbf{x}[351]} = 0.8795\ ,$$

so to obtain the correct measure of the intensity of the received RS sequence, sample 351 in signal (g) should be divided by this value to compensate for the effect of the linear transform.

In this chapter a general result on applying linear transforms to RS sequences is presented. The above example demonstrates a special case of this result. To get an impression on the motivation for believing in the existence of a relatively simple description of the phenomenon seen in Fig. 12.1(g), i.e. the seemingly well-ordered location of the possible affected samples, the result of applying the polynomial denoising to not just a single RS sequence, but to the entire RST is shown in Fig. B.1, B.2 and B.3 in Appendix B on page 317–319. The matrices shown are on the form $\mathbf{P}^{(N)}\mathbf{L}\mathbf{P}^{(N)}$ where $\mathbf{L}$ is a block

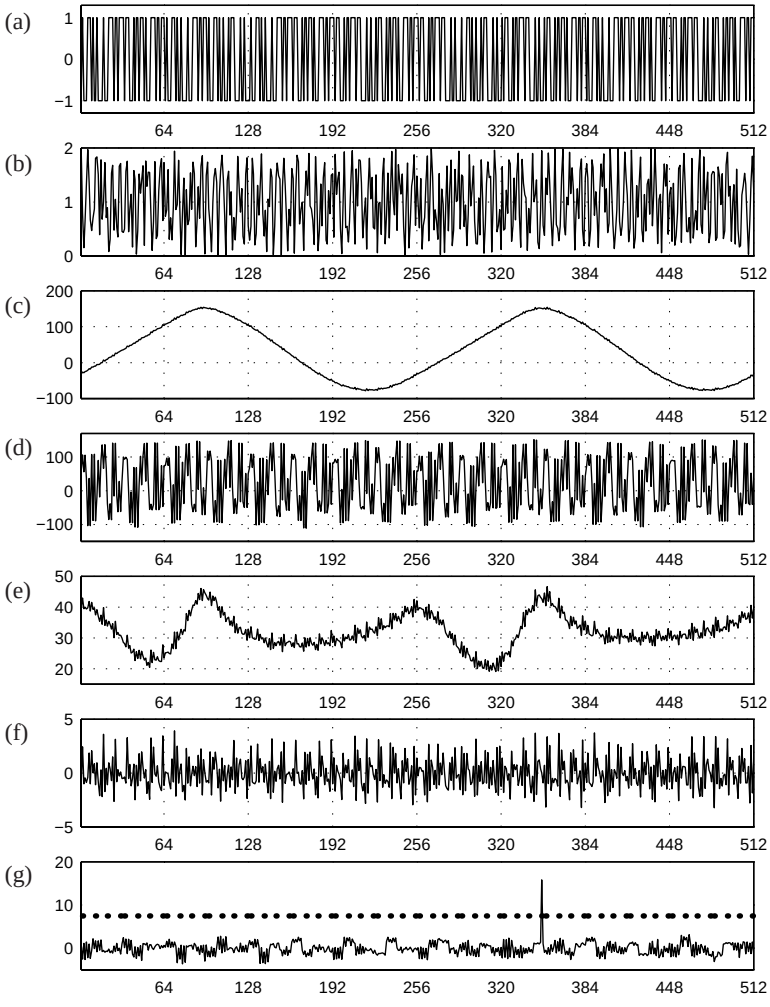


Figure 12.1: From top to bottom: (a) is the RST of a 512 sample signal with all but sample number 351 vanishing. (b) is the power spectrum of the RS sequence. (c) shows the unprocessed, received signal. (d) is the RST of (c). (e) shows the received signal with the most dominant sinusoid removed. (f) is the received signal subjected to a low-degree polynomial removal procedure. (g) is the RST of (f). The dots in (g) show the location of non-vanishing entries in $\mathbf{P}^{(9)}\mathbf{Lx}$.

diagonal matrix with 2^n blocks that remove polynomial content of degree m. The small black squares show where the matrix in non-vanishing. The gray checkerboard pattern in the matrices helps to identify the pattern. Note that the size of the gray checkers equals the number of blocks in the diagonal transform. The same matrix is shown in Fig. B.4 and B.5 where the blocks in the **L** is the periodized WT matrix with the Symlets 8 filter. The latter figure actually shows the matrix itself rather than where it is non-vanishing.

12.2 Self-Similarity Properties of the RST

The Rudin-Shapiro transform has an inherited dyadic structure which is evident by inspection of the definition laid out in (11.16), (11.17), and (11.18). Some consequences of this structure was also discussed in Section 11.4.3. The following definition describes how the linear transform in general is applied in a manner that complies with this structure.

Definition 12.1 (Dyadic Linear Transform)
Let $k, J \in \mathbb{N}$, $k < J$. Let $\mathcal{L}^{(k)} : \mathbb{R}^{2^k} \mapsto \mathbb{R}^{2^k}$ be a linear transforms and define

$$\mathcal{L}^{(J,k)} = \mathbf{I}^{(J-k)} \otimes \mathcal{L}^{(k)} ,$$

where $\mathbf{I}^{(J-k)}$ is the unity matrix of dimensions $2^{J-k} \times 2^{J-k}$.

Note that $\otimes$ means Kronecker product. This definition shows that $\mathcal{L}^{(J,k)}$ applies the same linear transform $\mathcal{L}^{(k)}$ to various parts of a $2^J \times 2^J$ matrix in such a way that the entire matrix is subjected to $\mathcal{L}^{(k)}$. Note that both dimensions and first entry of the submatrices are powers of 2. This in turn means that the result of applying the linear transform is to some extend governed by the self-similarity of the RST.

The appending rule in (11.5) as well as the product in (11.18) reveals that an RST matrix $\mathbf{P}^{(J)}$ is constructed by means of the one step smaller RST matrix $\mathbf{P}^{(J-1)}$. The logical extreme of this observation is that $\mathbf{P}^{(J)}$ depends on $\mathbf{P}^{(s)}$ for any $s < J$. The exact nature of this dependency is described in this section.

But first a few word on notation: The mth row of a matrix $\mathbf{A}$ is denoted $\mathbf{a}_m$, and the (m, n) entry is denoted $\mathbf{A}[m, n]$. The row and column count starts at zero. The $m \times n$ and $m \times m$ matrices of all 1's is denoted by $\mathbf{1}_{m\times n}$ and $\mathbf{1}_m$, respectively. The symbols $\odot$ and $\otimes$ means entry-by-entry multiplication and Kronecker product, respectively.

The self-similar structure of the RST is essentially due to the recursiveness of the original definition. This recursiveness is basically described by a starting point and the recursive rule. The following definition pinpoints these two elements in the RST matrix. The 'starting point' is given by $\mathbf{\Pi}_0$ and $\mathbf{\Pi}_1$, while the recursive rule is given by $\hat{\mathbf{\Pi}}$. This is confirmed in Lemma 12.3.

Definition 12.2
Let $2^{-J/2}\mathbf{P}^{(N)}$ *be an RST matrix, and let* $s \in \mathbb{N}$ *be such that* $J \geq s+1$*, and let* $K = 2^s$*. Define the two* $2^J \times 2^s$ *matrices*

$$\mathbf{\Pi}_0 = \left[\mathbf{p}_0^{(N)} \quad \cdots \quad \mathbf{p}_{K-1}^{(N)}\right] \odot \left(\mathbf{p}_0^{(N)} \otimes \mathbf{1}_{1\times K}\right)$$
$$\mathbf{\Pi}_1 = \left[\mathbf{p}_K^{(N)} \quad \cdots \quad \mathbf{p}_{2K-1}^{(N)}\right] \odot \left(\mathbf{p}_K^{(N)} \otimes \mathbf{1}_{1\times K}\right).$$

Define also the $2^J \times 2^{J-s}$ *matrix*

$$\hat{\mathbf{\Pi}} = \left[\mathbf{p}_0^{(N)} \quad \mathbf{p}_K^{(N)} \quad \mathbf{p}_{2K}^{(N)} \quad \cdots \quad \mathbf{p}_{(2^{J-s}-1)K}^{(N)}\right].$$

Define further for $J \geq s+2$ *four square matrices* $\hat{\mathbf{\Pi}}_0$ *through* $\hat{\mathbf{\Pi}}_3$ *such that*

$$\left[\hat{\mathbf{\Pi}}_0^\top \quad \hat{\mathbf{\Pi}}_1^\top \quad \hat{\mathbf{\Pi}}_2^\top \quad \hat{\mathbf{\Pi}}_3^\top\right]^\top$$

equals the first 2^{J-s+2} *rows of* $\hat{\mathbf{\Pi}}$*.*

Note that the $\mathbf{P}^{(N)}$ in the above definition has entries of unit size.

The matrix $\hat{\mathbf{\Pi}}$ is probably the most interesting matrix presented in this chapter. It is namely this matrix which indicates what samples in the denoised, transformed signals is affected by the denoising, and which are not. However, at this point it is no obvious how this can be, but it will be demonstrated later, once the proper frame has been established.

In this definition $\mathbf{\Pi}_0$ is the first 2^s columns of $\mathbf{P}^{(N)}$, and $\mathbf{\Pi}_1$ the subsequent 2^s columns of $\mathbf{P}^{(N)}$. A change of sign is applied on all rows of $\mathbf{\Pi}_0$ and $\mathbf{\Pi}_1$ such that the first column of both matrices are all 1's. Because of the self-similar structure of the RST matrix these columns can now be regarded as building blocks from which the entire RST matrix can be constructed. This is stated in the following lemma.

Lemma 12.3
Let $2^{-J/2}\mathbf{P}^{(N)}$ *be an RST matrix, and let* s*,* $\mathbf{\Pi}_0$*,* $\mathbf{\Pi}_1$*, and* $\hat{\mathbf{\Pi}}$ *be as in Definition 12.2. Then*

$$\mathbf{P}^{(N)} = \left(\mathbf{1}_{1\times 2^{J-s-1}} \otimes \begin{bmatrix}\mathbf{\Pi}_0 & \mathbf{\Pi}_1\end{bmatrix}\right) \odot \left(\hat{\mathbf{\Pi}} \otimes \mathbf{1}_{1\times 2^s}\right)$$

Proof
The lemma follows immediately from the appending rule demonstrated in (11.5) used to construct the $\mathbf{P}^{(N)}$ matrix. □

The main interest in this chapter is examining the consequences of applying a block diagonal linear transform to an RS sequence prior to transforming it. All RS sequences can be included by regarding the matrix $\mathbf{P}^{(N)}\mathcal{L}^{(J,s)}\mathbf{P}^{(N)}$. The remaining part of this chapter is dedicated to investigating how the block diagonal structure interferes with the RST on both sides. Since the inference has a block structure with blocks of size 2^k and the RST

can be regarded as constructed of size 2^k submatrices (which are really RSTs put together as described by Lemma 12.3) an important tool in the investigation is four permutation matrices which defines the relation between all the various size 2^k submatrices of the size 2^J RST.

The believe that such a relation exists is supported by the matrices shown in Section B.1 through B.5. Evidently, a number of permutation matrices arises when some linear transforms are applied in the block diagonal way explained in the beginning of this chapter. These matrices can be composed of the four different matrices shown in the first four rows in Fig. B.6. It is also obvious from Fig. B.5 that the exact (quantitatively) nature of relation is a priori not easily identifiable.

Before investigating this further the permutation matrices must be defined.

Definition 12.4 (Permutation Matrices)
Define $\mathbf{I}^{(N)}$ as the unity matrix of size $2^J \times 2^J$, and $\hat{\mathbf{I}}^{(N)}$ as the anti-identity matrix of the same size. Define then

$$\begin{aligned}
\tilde{\mathbf{I}}^{(J)} &= \hat{\mathbf{I}}^{(J-1)} \otimes \begin{bmatrix} 0 & (-1)^{J+1} \\ (-1)^J & 0 \end{bmatrix}, \\
\mathbf{T}^{(J)} &= \mathbf{T}^{(J-1)} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^{J-1}, \qquad \mathbf{T}^{(0)} = 1\,, \\
\boldsymbol{\Upsilon}^{(J)} &= \mathbf{T}^{(J-1)} \otimes \left(\begin{bmatrix} 0 & 1 \\ (-1)^J & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^J \right) \\
\tilde{\boldsymbol{\Upsilon}}^{(J)} &= -\tilde{\mathbf{I}}^{(J)} \boldsymbol{\Upsilon}^{(J)}\,.
\end{aligned}$$

Note that $\mathbf{T}$ is just an auxiliary matrix used to construct $\boldsymbol{\Upsilon}$. These matrices have a very simple structure. In particular, they are permutation matrices, i.e. they interchange rows (or columns) when applied to another matrix. Examples for size 2^5 and 2^6 are given in Fig. 12.2. A more detailed examples which includes the intermediate matrices in the construction is given in Fig. B.6 in Appendix B. These four matrices grasps the differences between the submatrices of the RST. The nature of these difference are directly responsible for the fact that some samples are altered by the block diagonal linear transform, and others are not. The evidence of the importance of these matrices are given by the following theorem. It states that by combining the four permutation matrices with the first quarter of $\hat{\boldsymbol{\Pi}}$, which is in itself an RST, the matrix $\hat{\boldsymbol{\Pi}}$ can be reconstructed.

The intimate relation between the possible impact of the block diagonal linear transform and the four permutation matrices means that, effectively, Theorem 12.5 states that applying a block diagonal linear transform to an RST transformed signal prior to inverse RST is essentially described by the two matrices $\mathbf{I}$ and $\boldsymbol{\Upsilon}$.

Theorem 12.5
Let $2^{-s/2}\mathbf{P}^{(s)}$ and $2^{-J/2}\mathbf{P}^{(N)}$ be two RST matrices, and let s and $\hat{\boldsymbol{\Pi}}$ be as in Defini-

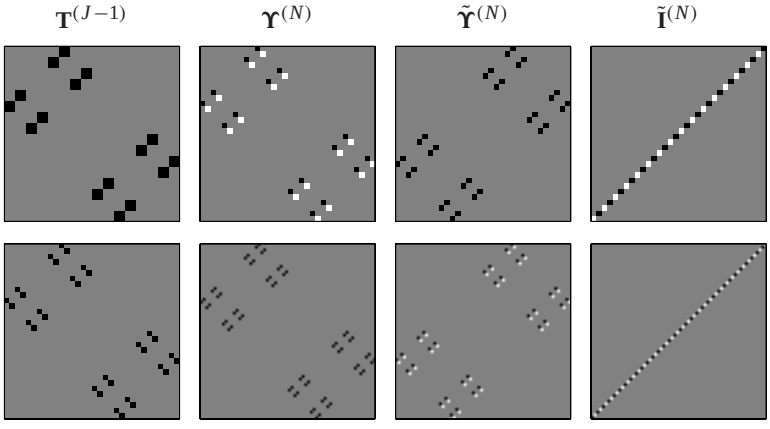


Figure 12.2: Examples of the $\mathbf{T}$, $\boldsymbol{\Upsilon}$, $\tilde{\boldsymbol{\Upsilon}}$, and $\tilde{\mathbf{I}}$ matrices for $J = 5$ and $J = 6$. (Black $= 1$, grey $= 0$, white $= -1$.)

tion 12.2. Let $K = 2^{J-s}$. *Then* $\hat{\mathbf{\Pi}}_0 = \sqrt{K}\mathbf{P}^{(J-s)}$, *and*

$$\hat{\mathbf{\Pi}} = \left(\mathbf{1}_{2^{s-2}\times 1} \otimes \left(\begin{bmatrix} \mathbf{I}^{(J-s)} \\ \boldsymbol{\Upsilon}^{(J-s)} \\ \tilde{\boldsymbol{\Upsilon}}^{(J-s)} \\ \tilde{\mathbf{I}}^{(J-s)} \end{bmatrix} \hat{\mathbf{\Pi}}_0\right)\right) \odot \left(\mathbf{p}_0^{(s)} \otimes \mathbf{1}_K\right). \tag{12.1}$$

Proof

The entries of $\hat{\mathbf{\Pi}}$ are given by (11.19), where $m = 0, \ldots, 2^J - 1$ and $n = 0, 2^s, 2 \cdot 2^s, \ldots, (K-1)2^s$, that is $n_r = 0$ for $r < s$. Therefore $\hat{\mathbf{\Pi}}[m, n] = f(m)g(m, n)$ with

$$f(m) = \prod_{j=J-s+1}^{J-1} (-1)^{m_j m_{j+1}}$$

and

$$g(m, n) = \prod_{j=0}^{J-s} (-1)^{(m_j + n_{J-s-j+1})(m_{j+1} + n_{J-s-j-1})} ,$$

where n has been reset to $n = 0, \ldots, 2^{J-s} - 1$. Since f does not depend on the $J - s$ LSBs of the binary representation of m, it is constant for $m = rK, \ldots, (r+1)K - 1$, for each $r = 0, \ldots, 2^s - 1$. Since g does not depend on the $s - 2$ MSBs of m if follows that

$$g(m, n) = g(m + 4K, n), \quad m = 0, \ldots, 2^J - 4K - 1 .$$

Consequently, the matrix consisting of the first K rows of $\hat{\mathbf{\Pi}}$ (in Definition 12.2 denoted $\hat{\mathbf{\Pi}}_0$) equals the 2^{s-2} matrices consisting of the rows $4rK$ through $(4r+1)K$, for $r =$

$0, \ldots, 2^{s-2} - 1$, except for a possible change of sign determined by $f(m)$. The same applies to $\hat{\mathbf{\Pi}}_1$, $\hat{\mathbf{\Pi}}_2$, and $\hat{\mathbf{\Pi}}_3$. Now, since $\hat{\mathbf{\Pi}}_0$ is the first 2^{J-s} rows of $\hat{\mathbf{\Pi}}$ its entries are given by $\hat{\mathbf{\Pi}}_0[m, n] = g(m, n)$ which is a scaled RST matrix of size $K \times K$, where the numerical value of the entries is 1. It now remains to show that

$$\hat{\mathbf{\Pi}}_1 = \mathbf{\Upsilon}^{(J-s)}\hat{\mathbf{\Pi}}_0, \quad \hat{\mathbf{\Pi}}_2 = \tilde{\mathbf{\Upsilon}}^{(J-s)}\hat{\mathbf{\Pi}}_0, \quad \hat{\mathbf{\Pi}}_3 = \tilde{\mathbf{I}}^{(J-s)}\hat{\mathbf{\Pi}}_0 \ .$$

The RST matrix is constructed recursively in such a manner that the $\hat{\mathbf{\Pi}}_0$, which originates in $\mathbf{P}^{(N)}$, depends solely on every 2^s elements of the first 2^{J-s-1} rows of $\mathbf{P}^{(J-1)}$, which in turn depends solely on every 2^s elements of the first first 2^{J-s-2} rows of $\mathbf{P}^{(J-2)}$, and so on. Ultimately, $\hat{\mathbf{\Pi}}_0$ depends solely on

$$\hat{\mathbf{\Pi}}_0 : \quad \begin{bmatrix} \mathbf{P}^{(s+1)}[0,0] & \mathbf{P}^{(s+1)}[0,2^s] \\ \mathbf{P}^{(s+1)}[1,0] & \mathbf{P}^{(s+1)}[1,2^s] \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

where the equality follows directly from (11.19). The same back tracing shows that $\hat{\mathbf{\Pi}}_1$, $\hat{\mathbf{\Pi}}_2$, and $\hat{\mathbf{\Pi}}_3$ depends solely on

$$\begin{aligned}
\hat{\mathbf{\Pi}}_1 : & \quad \begin{bmatrix} \mathbf{P}^{(s+1)}[2,0] & \mathbf{P}^{(s+1)}[2,2^s] \\ \mathbf{P}^{(s+1)}[3,0] & \mathbf{P}^{(s+1)}[3,2^s] \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \\
\hat{\mathbf{\Pi}}_2 : & \quad \begin{bmatrix} \mathbf{P}^{(s+1)}[4,0] & \mathbf{P}^{(s+1)}[4,2^s] \\ \mathbf{P}^{(s+1)}[5,0] & \mathbf{P}^{(s+1)}[5,2^s] \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \\
\hat{\mathbf{\Pi}}_3 : & \quad \begin{bmatrix} \mathbf{P}^{(s+1)}[6,0] & \mathbf{P}^{(s+1)}[6,2^s] \\ \mathbf{P}^{(s+1)}[7,0] & \mathbf{P}^{(s+1)}[7,2^s] \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}.
\end{aligned}$$

Consequently, the four matrices $\hat{\mathbf{\Pi}}_1$, $\hat{\mathbf{\Pi}}_2$, and $\hat{\mathbf{\Pi}}_3$ can be constructed by following the appending rule laid out in (11.12) but with varying start conditions

$$\begin{aligned}
\hat{\mathbf{\Pi}}_1 : P_{1,0} = Q_{1,1} = \ & 1 + e^{i2\pi\xi} && P_{1,1} = Q_{1,0} = -1 + e^{i2\pi\xi}, \\
\hat{\mathbf{\Pi}}_2 : P_{1,0} = Q_{1,1} = \ & 1 - e^{i2\pi\xi} && P_{1,1} = Q_{1,0} = \ 1 + e^{i2\pi\xi}, \\
\hat{\mathbf{\Pi}}_3 : P_{1,0} = Q_{1,1} = & -1 + e^{i2\pi\xi} && P_{1,1} = Q_{1,0} = \ 1 + e^{i2\pi\xi}.
\end{aligned}$$

The difference between the starting conditions of $\hat{\mathbf{\Pi}}_0$ and $\hat{\mathbf{\Pi}}_1$ is sign of $P_{1,1}$. Examining ($\mathbf{p}$ is now rows of the RST matrix)

$$\mathbf{P}^{(2)} = \begin{bmatrix} \mathbf{p}_0^{(2)} \\ \mathbf{p}_1^{(2)} \\ \mathbf{p}_2^{(2)} \\ \mathbf{p}_3^{(2)} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}^{(2)} = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 \\ \mathbf{p}_0 & -\mathbf{p}_1 \\ \mathbf{p}_1 & \mathbf{p}_0 \\ -\mathbf{p}_1 & \mathbf{p}_0 \end{bmatrix}^{(1)}$$

it is obvious that changing the sign of $\mathbf{p}_1^{(1)}$ is equivalent to exchange the first and second rows, and the third and fourth rows of $\mathbf{P}^{(2)}$. (The second equality shows how the size

indicator is moved outside the matrix for convenience, and applies to all entries of the matrix.) Consequently, the 4×4 version of the matrix $\hat{\mathbf{\Pi}}_1$ equals $\mathbf{B}\hat{\mathbf{\Pi}}_0$, where

$$\mathbf{B} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \mathbf{\Upsilon}^{(2)} .$$

The next step is the 8×8 RST matrix

$$\mathbf{P}^{(3)} = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 \\ \mathbf{p}_0 & -\mathbf{p}_1 \\ \mathbf{p}_1 & \mathbf{p}_0 \\ -\mathbf{p}_1 & \mathbf{p}_0 \\ \mathbf{p}_2 & \mathbf{p}_3 \\ \mathbf{p}_2 & -\mathbf{p}_3 \\ \mathbf{p}_3 & \mathbf{p}_2 \\ -\mathbf{p}_3 & \mathbf{p}_2 \end{bmatrix}^{(2)} = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \mathbf{p}_0 & -\mathbf{p}_1 \\ \mathbf{p}_0 & \mathbf{p}_1 & -\mathbf{p}_0 & \mathbf{p}_1 \\ \mathbf{p}_0 & -\mathbf{p}_1 & \mathbf{p}_0 & \mathbf{p}_1 \\ -\mathbf{p}_0 & \mathbf{p}_1 & \mathbf{p}_0 & \mathbf{p}_1 \\ \mathbf{p}_1 & \mathbf{p}_0 & -\mathbf{p}_1 & \mathbf{p}_0 \\ \mathbf{p}_1 & \mathbf{p}_0 & \mathbf{p}_1 & -\mathbf{p}_0 \\ -\mathbf{p}_1 & \mathbf{p}_0 & \mathbf{p}_1 & \mathbf{p}_0 \\ \mathbf{p}_1 & -\mathbf{p}_0 & \mathbf{p}_1 & \mathbf{p}_0 \end{bmatrix}^{(1)} \tag{12.2}$$

Exchanging $\mathbf{p}_0^{(2)}$ and $\mathbf{p}_1^{(2)}$ (the consequence of changing the sign of $\mathbf{p}_1^{(1)}$) is equivalent to exchanging the first and the second rows with the third and the fourth rows of $\mathbf{P}^{(2)}$ and changing the sign of the second and fourth rows. Likewise the fifth and the sixth rows is exchanged with the seventh and the eighth rows. This is not incidental but a direct consequence of (11.12). Interchanging one pair of rows with another pair in this fashion and changing signs of the even rows is accomplished by exchanging every non-vanishing and every vanishing entry in $\mathbf{\Upsilon}^{(2)}$ with

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

respectively. Thus the 8×8 matrix $\hat{\mathbf{\Pi}}_1$ equals $\mathbf{B}\hat{\mathbf{\Pi}}_0$, where

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} = \mathbf{\Upsilon}^{(3)} .$$

Now, when constructing $\mathbf{P}^{(4)}$ in exactly the same fashion as (12.2), the changing of the sign of $\mathbf{p}_1^{(3)}$ (and the other odd-indexed $\mathbf{p}^{(3)}$) will result in the same considerations as the

ones leading to interchanging consecutive rows in $\mathbf{P}^{(2)}$. Thus, by replacing non-vanishing and vanishing entries in $\boldsymbol{\Upsilon}^{(3)}$ with

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

respectively, the 16×16 permutation matrix for transforming $\hat{\mathbf{\Pi}}_0$ to $\hat{\mathbf{\Pi}}_1$ is constructed. That, of course, is $\boldsymbol{\Upsilon}^{(4)}$.

Similar arguments leads to $\hat{\mathbf{\Pi}}_0 = \tilde{\boldsymbol{\Upsilon}}^{(J)}\hat{\mathbf{\Pi}}_2$ and $\hat{\mathbf{\Pi}}_0 = \tilde{\mathbf{I}}^{(J)}\hat{\mathbf{\Pi}}_3$. □

Having established the applicability of the permutation matrices the following corollary shows how to apply the previous result to obtain a qualitative description of the impact of applying the block diagonal linear transform.

Corollary 12.5.1
For a given J let s and $\hat{\mathbf{\Pi}}$ be as in Definition 12.2. Let

$$\mathbf{A} = \left(\mathbf{I}^{(2)} \otimes \mathbf{I}^{(J-s)}\right) + \left(\boldsymbol{\Upsilon}^{(2)} \otimes \boldsymbol{\Upsilon}^{(J-s)}\right) + \left(\tilde{\boldsymbol{\Upsilon}}^{(2)} \otimes \tilde{\boldsymbol{\Upsilon}}^{(J-s)}\right) + \left(\tilde{\mathbf{I}}^{(2)} \otimes \tilde{\mathbf{I}}^{(J-s)}\right).$$

Then

$$\hat{\mathbf{\Pi}}\hat{\mathbf{\Pi}}^{\top} = \left(\mathbf{1}_{2^{s-2}} \otimes \mathbf{A}\right) \odot \left(\mathbf{p}_0^{(s)}(\mathbf{p}_0^{(s)})^{\top} \otimes \mathbf{1}_{2^{J-s}}\right). \tag{12.3}$$

Proof
Except for $\mathbf{A}$ all matrices in this proof have size $2^{J-s} \times 2^{J-s}$. First note that

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & \boldsymbol{\Upsilon} & \tilde{\boldsymbol{\Upsilon}} & -\tilde{\mathbf{I}} \\ \boldsymbol{\Upsilon} & \mathbf{I} & \tilde{\mathbf{I}} & -\tilde{\boldsymbol{\Upsilon}} \\ \tilde{\boldsymbol{\Upsilon}} & -\tilde{\mathbf{I}} & \mathbf{I} & \boldsymbol{\Upsilon} \\ \tilde{\mathbf{I}} & -\tilde{\boldsymbol{\Upsilon}} & \boldsymbol{\Upsilon} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{\Pi}}_0 \\ \hat{\mathbf{\Pi}}_1 \\ \hat{\mathbf{\Pi}}_2 \\ \hat{\mathbf{\Pi}}_3 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{\Pi}}_0^{\top} & \hat{\mathbf{\Pi}}_1^{\top} & \hat{\mathbf{\Pi}}_2^{\top} & \hat{\mathbf{\Pi}}_3^{\top} \end{bmatrix}$$

The first equality follows immediately from Definition 12.4. The second equality follows from the facts that $\boldsymbol{\Upsilon}$ and $\tilde{\boldsymbol{\Upsilon}}$ are orthogonal and symmetric, that $\tilde{\mathbf{I}}$ are orthogonal, and that $\tilde{\mathbf{I}}^{\top} = -\tilde{\mathbf{I}}$. For instance

$$\hat{\mathbf{\Pi}}_2\hat{\mathbf{\Pi}}_3^{\top} = \tilde{\boldsymbol{\Upsilon}}\hat{\mathbf{\Pi}}_0(\tilde{\mathbf{I}}\,\hat{\mathbf{\Pi}}_0)^{\top} = \tilde{\boldsymbol{\Upsilon}}\tilde{\mathbf{I}}^{\top} = \left(-\tilde{\mathbf{I}}\boldsymbol{\Upsilon}\right)^{\top}\tilde{\mathbf{I}}^{\top} = \boldsymbol{\Upsilon}$$

and

$$\hat{\mathbf{\Pi}}_3\hat{\mathbf{\Pi}}_1^{\top} = \tilde{\mathbf{I}}\,\hat{\mathbf{\Pi}}_0\hat{\mathbf{\Pi}}_0\boldsymbol{\Upsilon} = \tilde{\mathbf{I}}\boldsymbol{\Upsilon} = -\tilde{\boldsymbol{\Upsilon}}\ .$$

Now (12.3) follows from (12.1). □

It was stated earlier in this chapter that $\hat{\mathbf{\Pi}}$ was an interesting matrix from an applicational point of view. In (12.3) it is seen that the 'outer product' of the matrix can be constructed by concatenating $\mathbf{A}$ matrices vertically and horizontally. Note that $\left(\mathbf{p}_0^{(s)}(\mathbf{p}_0^{(s)})^{\top} \otimes \mathbf{1}_{2^{J-s}}\right)$ is just appropriate changes of signs of submatrices of $\left(\mathbf{1}_{2^{s-2}} \otimes \mathbf{A}\right)$. The $\mathbf{A}$ matrix is

constructed as a mix of the four matrices that describes the relation between the various submatrices of size 2^{J-s} of the RST matrix, and is thus a 'worst case' of what can happen when a block diagonal linear transform is applied. The linear transform itself is not involved in $\mathbf{A}$ and therefore it only qualifies which entries in the RST can be affected, but obviously does not quantify the effect as this depends on the choice of linear transform.

In short, the non-vanishing entries of $\left(\mathbf{1}_{2^{s-2}} \otimes \mathbf{A}\right)$, and thus also of $\hat{\mathbf{\Pi}}\hat{\mathbf{\Pi}}^{\top}$, are the entries of $\mathbf{P}^{(N)}\boldsymbol{\mathcal{L}}^{(J,s)}\mathbf{P}^{(N)}$ which are potentially non-vanishing. In most cases the latter matrix will indeed have entries which are vanishing although the corresponding entries in the former matrices are not. The $\hat{\mathbf{\Pi}}\hat{\mathbf{\Pi}}^{\top}$ for the eight times eight polynomial denoising in Chapter 5 is shown in Fig. 4.9 on page 72. The corresponding $\mathbf{P}^{(N)}\boldsymbol{\mathcal{L}}^{(J,s)}\mathbf{P}^{(N)}$ is shown in the same figure.

In order to present the final theorem of this chapter one more matrix is needed. Define the $2^s \times 2^{s+1}$ diagonal matrices

$$\mathbf{S}_u[m,n] = \begin{cases} (-1)^{un_s+um_1+\sum_{k=1}^{s-1} m_k m_{k+1}} & \text{for } m = n \mod 2^s \\ 0 & \text{otherwise} , \end{cases}$$

where m_x is the binary representation of m, with m_1 being the LSB.

Theorem 12.6
For a given J let s and $\mathbf{\Pi}_0$ be as in Definition 12.2, and let $\boldsymbol{\mathcal{L}}^{(J,s)}$ be a linear transformation as in Definition 12.1. Then

$$\begin{aligned}\mathbf{P}^{(N)}\boldsymbol{\mathcal{L}}^{(J,s)}\mathbf{P}^{(N)} = \mathbf{K}_0^{(s-1)} \otimes (\mathbf{I}+\mathbf{\Upsilon}) + \mathbf{K}_1^{(s-1)} \otimes (\mathbf{I}-\mathbf{\Upsilon}) \\ + \tilde{\mathbf{K}}_0^{(s-1)} \otimes (\tilde{\mathbf{I}}-\tilde{\mathbf{\Upsilon}}) + \tilde{\mathbf{K}}_1^{(s-1)} \otimes (\tilde{\mathbf{I}}+\tilde{\mathbf{\Upsilon}}) ,\end{aligned}$$

where $\mathbf{I}$, $\mathbf{\Upsilon}$, $\tilde{\mathbf{\Upsilon}}$, and $\tilde{\mathbf{I}}$ all have size 2^{J-s+1}, and with

$$\begin{aligned}\mathbf{K}_0^{(s-1)} &= \mathbf{S}_0\boldsymbol{\mathcal{L}}'^{(s)}\mathbf{S}_0^{\top} \\ \mathbf{K}_1^{(s-1)} &= \mathbf{S}_1\boldsymbol{\mathcal{L}}'^{(s)}\mathbf{S}_1^{\top} \\ \tilde{\mathbf{K}}_0^{(s-1)} &= \mathbf{S}_1\boldsymbol{\mathcal{L}}'^{(s)}\mathbf{S}_0^{\top} \\ \tilde{\mathbf{K}}_1^{(s-1)} &= \mathbf{S}_0\boldsymbol{\mathcal{L}}'^{(s)}\mathbf{S}_1^{\top}\end{aligned}$$

where

$$\boldsymbol{\mathcal{L}}'^{(s)} = \begin{bmatrix}\mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}\mathbf{\Upsilon}^{(s-1)}\end{bmatrix}\boldsymbol{\mathcal{L}}^{(s)}\begin{bmatrix}\mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}\mathbf{\Upsilon}^{(s-1)}\end{bmatrix}^{\top} .$$

The following proof is rather sketchy as a number of details is still not properly described. Hopefully, the following description gives an idea of how the full proof might go.

Proof
First note that there exists an $2^s \times 2^s$ matrix $\mathbf{V}$ such that

$$\mathbf{P}^{(N)}(\mathbf{K} \otimes \mathbf{U})\mathbf{P}^{(N)} = \mathbf{V} \otimes \mathbf{I}^{(J-s)}$$

for a fixed choice of

$$\begin{aligned} &\mathbf{K} \in \{\mathbf{K}_0^{(s-1)}, \mathbf{K}_1^{(s-1)}, \tilde{\mathbf{K}}_0^{(s-1)}, \tilde{\mathbf{K}}_1^{(s-1)}\}, \\ \text{and} \quad &\mathbf{U} \in \{\mathbf{I}^{(J-s+1)}, \boldsymbol{\Upsilon}^{(J-s+1)}, \tilde{\boldsymbol{\Upsilon}}^{(J-s+1)}, \tilde{\mathbf{I}}^{(J-s+1)}\}. \end{aligned}$$

Writing $\mathbf{K}$ as

$$\sum_{m=0}^{2^{s-1}-1} \sum_{n=0}^{2^{s-1}-1} \mathbf{E}_{m,n}$$

where $\mathbf{E}_{m,n}$ is the canonical matrix with 1 at the (m, n) entry and 0 otherwise. Then

$$\mathbf{P}^{(N)}(\mathbf{K} \otimes \mathbf{U})\mathbf{P}^{(N)} = \sum_{m=0}^{2^{s-1}-1} \sum_{n=0}^{2^{s-1}-1} \left[\mathbf{P}^{(N)}(\mathbf{E}_{m,n} \otimes \mathbf{U})\mathbf{P}^{(N)}\right].$$

The matrix $\mathbf{P}^{(N)}(\mathbf{E}_{m,n} \otimes \mathbf{U})\mathbf{P}^{(N)}$ is equal to the outer product of the submatrix $\boldsymbol{\Lambda}_0$ consisting of rows $m2^{J-s+1}$ through $(m+1)2^{J-s+1}-1$ and submatrix $\boldsymbol{\Lambda}_1$ consisting of rows $n2^{J-s+1}$ through $(n+1)2^{J-s+1}-1$ of $\mathbf{P}^{(N)}$, the latter permuted in order according to $\mathbf{U}$. That is,

$$\mathbf{P}^{(N)}(\mathbf{E}_{m,n} \otimes \mathbf{U})\mathbf{P}^{(N)} = \boldsymbol{\Lambda}_0 \mathbf{U} \boldsymbol{\Lambda}_1^\top$$

with

$$\begin{aligned} \boldsymbol{\Lambda}_0 &= \left[\mathbf{p}_{m2^{J-s+1}}^{(J)} \quad \cdots \quad \mathbf{p}_{(m+1)2^{J-s+1}-1}^{(J)}\right] \\ \boldsymbol{\Lambda}_1 &= \left[\mathbf{p}_{n2^{J-s+1}}^{(J)} \quad \cdots \quad \mathbf{p}_{(n+1)2^{J-s+1}-1}^{(J)}\right], \end{aligned}$$

and according to (11.19) the rows $k2^{s-1}$ through $(k+1)2^{s-1}$ of $\boldsymbol{\Lambda}_0$ and $\mathbf{U}\boldsymbol{\Lambda}_1$, for fixed $k = 0, 1, \ldots, 2^{J-s+1}$, are identical up to change of sign.

The appending rule laid out in (11.12) shows that the RST can at any dyadic level be regarded as a matrix consisting of rows of alternating and concatenated $\mathbf{p}$ and $\mathbf{q}$ vectors. Choosing the level at which these vectors have length 2^{J-s+1} the $\boldsymbol{\Lambda}_0$ is either a "$\mathbf{p}$-matrix" or a "$\mathbf{q}$-matrix". The same goes for $\boldsymbol{\Lambda}_1$.

Define now $\tilde{\boldsymbol{\Lambda}}_0$ as the matrix consisting of every 2^{s-1}th row of $\boldsymbol{\Lambda}_0$, and define $\tilde{\boldsymbol{\Lambda}}_1$ in the same way. Then $\tilde{\boldsymbol{\Lambda}}_0$ is unitary and $\tilde{\boldsymbol{\Lambda}}_0 = (\mathbf{W}_{m+n} \otimes \mathbf{I}^{(J-s)})\tilde{\boldsymbol{\Lambda}}_1$ where

$$\mathbf{W}_x = \begin{cases} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & \text{for } x \bmod 4 = 0 \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} & \text{for } x \bmod 4 = 1 \\ \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} & \text{for } x \bmod 4 = 2 \\ \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} & \text{for } x \bmod 4 = 3\,. \end{cases}$$

The first matrix corresponds to $\mathbf{\Lambda}_0$ and $\mathbf{\Lambda}_1$ being both $\mathbf{p}$ or $\mathbf{q}$, and likewise for the third matrix. Now, note the relation between $\mathbf{p}$ and $\mathbf{q}$ shown in (11.27). It follows that the second matrix corresponds to $\mathbf{\Lambda}_0$ begin a $\mathbf{p}$-matrix and $\mathbf{\Lambda}_1$ a $\mathbf{q}$-matrix, or vice versa. Likewise for the fourth matrix. The change of sign in the third and fourth matrix follows from (11.19). □

Discussions and Future Work

13

13.1 Robust Channel Gain Measurements (Part I)

The first part of the thesis presented a signal processing algorithm for improving on a number of parameters in active sensors. A list of interesting parameters was given in Section 3.5, and it was discussed which of these are most important in relation to common problems in existing sensors. It was argued that robustness and low-cost are two key parameters, and the presented algorithm thus aimed at providing robustness without using anything but simple and stable signal processing methods capable of real-time execution in low-cost hardware. The basic idea was to use invertible transforms and to exploit the full potential of multiplexing given by these transforms to obtain estimates of the transmission conditions.

The entire algorithm was presented in Chapter 4 in considerable details. The presentation was of mixed mathematical and engineering nature in the sense that mathematically founded solutions were applied to specific problems that were either a priori identified or had arisen in the test setups.

No alternatives to the suggested algorithm has been discussed in the thesis. Obviously, there exists a number of different analog implementation of active sensors, but these are not considered to be alternatives to a digital implementation (see Section 3.3). The list of digital alternatives is rather short as the author has not been able to find any previous research result in the area of signal processing in low-cost active sensors. Admittedly, the fact that the presented algorithm has fulfilled the criteria given in the beginning of Part I has not encouraged the author to a vigorous and extensive search through literature for alternatives.

13.1.1 Experiments

To test the algorithm it was applied to four different test setups. These were all based on infrared technology, but with different implementations of the electrical circuits. Two of the setups were based on existing products while the two others were made from scratch. The software implementation of the algorithm demonstrated the necessity and stability of the individual steps in the algorithm.
The estimation of channel gain is the ultimate purpose of the algorithm, and in Chapter 5

it was demonstrated that the algorithm is capable of providing such an estimate. This was the case not only case in low-noise conditions, but also when more severe noise was present. The estimate was based on correlation between the emitted and received signal (with a transform/inverse transform in between) because it is an easy signal processor operation and because it is mathematically the optimal operation in a random noise scenario.

The algorithm also demonstrated that when the noise condition are such that it is either time-consuming, computational expensive, or downright impossible to acquire a reasonable estimate it is possible to provide an alternative to the traditional sensor output 'detection/no detection'. This is due to the validation methods which based on relatively few computations are quite effective in deciding whether a given measurement fulfills an accuracy requirement or not. While it is possible to imagine other ways of exploiting the information from the noise-channels (like joint validation in system with multiple emitters, i.e. validating the three transmission channels in the second test setup by a set of linked hypotheses rather than three separate hypotheses) no other validation methods has been discussed. Partly because the presented methods works quite well, partly due to lack of time.

Some of the steps in the algorithm was dedicated to denoising. The need for such steps was clearly demonstrated with the recorded test signals. To that end Chapter 4 and 5 presented some denoising methods adapted to comply with the structure of the algorithm and the limited availability of computational power. While the presented methods were indeed successful in denoising the signal, other methods might easily be useful or indeed necessary in case of other types of noise. In particular, in a given environment some particular noise structure might be predominant in which case the presented methods would most likely fail. In this thesis only infrared technology has been tested, and it is reasonable to expect other technologies such as High-Frequency or acoustic waves will require somewhat different means in respect to denoising. However, the author do believe that the time and frequency-localized noise as well as random noise is predominant in most sensors systems, independently of type.

13.1.2 Algorithms or Hardware?

Throughout Part I of the thesis the main theme has been 'robustness by means of signal processing'. A large effort has been put into describing an algorithm with this property, and the functionality of the algorithm has been demonstrated by applying it to a series of test signals. There has been only brief comments and mentioning of the point of view that many of the problems addressed by the algorithm can be solved by means of simple hardware, i.e. that robustness can be achieve without the need for a signal processor. For instance, it can be argued that the sinusoid originating from the artificial laboratory lighting in the third test setup, see for example Fig. 5.17 on page 116, can be removed simple by added a plastic screening which is transparent only to infrared light combined with a proper transfer function of the amplifier, perhaps even with an analog high pass filter. The infrared transparent plastic screening is used in the BeoSound Ouverture to

create the effect of black side panels, compare Fig. 3.3, page 23, and Fig. 5.1, page 93. Transients can be also be handled in hardware, for instance by proper low pass filtering.

It has been stated a number of times in the thesis that the sensor market is huge, and that a majority of the existing sensors employ analog solutions. This obviously means that there exists quite a few sensors that work well, i.e. that do not experience failure or respond to occurrences they were designed to not respond to.

With these arguments in mind one might raise the question of whether it is indeed necessary to employ signal processing and on-chip computers in the strive for increased robustness. The author believes this to be the case, for the following reasons.

Time and frequency-localized noise can occur in all sensors While hardware of various kind can reduce the amplitude of noise it is rarely possible to completely eliminate it. A sufficiently powerful disturbance will 'penetrate' whatever filters have been used (such as optical and electrical), so while the SNR of the internal signal is much better than the external signal there might still be room for improving it further. And this is exactly what the algorithm presented in first part of the thesis is designed to do. Therefore, signal processing does not have to be an alternative to hardware denoising. The two methods might just as well supplement each other. In fact, hardware denoising is often necessary to avoid saturation of the ADC.

The test signals presented in this thesis might not have been generated under conditions which resemble those of commercial sensors, but as argued this does not mean that the presented noise-types does not occur in commercial sensors. The results in Chapter 5 of applying the algorithm to the test signals is therefore to be regarded as a demonstration of to what extent the algorithm can handle different types of noise, i.e. what improvement can be made of the SNR in given scenarios. It is *not* to be regarded as a demonstration of the algorithm handling noise generated by artificial lighting or a remote control. These disturbances have been used simply because they are readily available and the author does not have access to more sophisticated means of replicating disturbances as they appear in commercial sensors.

Frequency multiplexing sensors are sensitive to harmonic noise One of the more common types of signals in active sensors is harmonic signals. These are easily generated and processed by analog hardware. A sensor based on a particular frequency is obviously quite sensitive to any disturbance at that frequency. In this case no filtering can prevent the disturbance from reaching the processing part of the sensor. Since harmonic signals are quite common in the human environment such a sensor is a priori more sensitive than a sensor based on signals localized in some other domain which are less common in the environment. Such signals are more easily generated by signal processing means than by analog means.

Increased robustness is responding properly in bad-case scenarios The argument that states that most sensors work well throughout there life time misses the point of this thesis. The author acknowledges that analog sensors designed to operate in a specific

environment will work fine in this environment. For instance, when a sensor is expected to experience white noise of a certain amplitude it is not necessary to employ signal processing to construct a sensor with an error rate of, say, 10^{-7}. And the sensor will work according to specification as long as no unforeseen events occur. However, increased robustness is not about reducing the error rate in a best-case scenario, but in a bad-case scenario. For many sensors that bad-case scenario rarely occurs, but when it does, an 'unprepared' sensor will have a much higher risk of given a misleading output.

Sensors do fail The methods presented in the first part of this thesis are not (just) given because the author finds the field of signal processing in active sensors interesting. They are included because the author believes that they are capable of solving some problems which have indeed occurred in commercial sensors. The author is aware of a number of cases where sensors have failed because of lack of robustness, i.e. the failure is due to inability to handle particular disturbances. And the author believes that the presented algorithm would have been able to avoid such failures.

Hardware is a variable cost Solving noise problems by applying hardware might not always be the most cost-efficient solution. While hardware is a variable cost, i.e. the cost depends on the total volume, the designing of an algorithm is a one-time expense. For a sufficiently large volume of products the algorithmic solutions is cheaper. Obviously, the signal processor must be included in this equation, but the point is that it is not a priori given that hardware denoising is the best solution. It is also important to remember that less hardware makes the entire design process easier, and thus cheaper.

13.2 Spatial Position (Part II)

The concept of sensor using several CGMs to determine the position of an object in three dimensions were presented in Part II. The idea is to have a set of emitter/receiver pairs at different positions such that light emitted from a number of position is reflect onto the same variety of positions. This generates a series of CGMs which can be regarded as relative measures of distance to the object. The challenge is then to map the set of CGMs into a three dimensional position.

Two mapping methods were presented in Part II. The first method is a neural network which is trained on real reflection data, and the second method is a geometric modeling of a emitter/receiver setup. None of the methods have been sufficiently convincing to rule out the other method, or any other conceivable methods, for that matter. In both methods it is imperative to have measurements of the real reflection map, and therefore such a data set was also acquired in Part II.

To facilitate a more complex geometric modeling than the one presented in this thesis (this is a part of the future work) a model of the reflection map was also made. This was

evaluated by means of the measured reflection map.

The geometric modeling of a emitter/receiver setup was carried out in 3D, while the neural network and reflection map modeling was done in 2D. This is partly because the author believes that if the modeling work in 2D it will also (be possible to make it) work in 3D, partly because it is cumbersome to acquire a 3D reflection data set (the means for doing this in an reproducible manner was not available during the Ph.D. study).

13.2.1 Neural Network

The first method has been developed in an ad hoc manner to provide an indication of whether a neural network is a good solution to the mapping problem. A network has been created by means of the neural network toolbox in MATLAB and trained on simulated data. The result of this test is inconclusive in the sense that although it seems possible, under some conditions, to map CGMs to position by means of a neural network, some factors are still unknown. This includes computational complexity, stability, and adaptability. Also, the author is not familiar with the theory of neural networks and does not have the prerequisites for determining whether other types of networks would perform better. A more thorough investigation is needed to determine this.

13.2.2 Geometric Solution based on Intersections of Spheroids

The second method is a modeling based on the assumption that the three dimensional reflection map of an emitter/receiver pair consists of 'concentric' prolate spheroids. The CGMs corresponds to radii in such spheroids and the spatial position can therefore be determined by intersection of the spheroids. The resulting mapping from CGMs to spatial position is therefore given as the solution to a set of three spheroid equations (as three spheroids is sufficient to provide an unique spatial position). The analytical solution allows for a series of purely mathematical observations which lead to a description of the optimal position for the emitters and receivers. It is also possible to provide means for 'converting' redundancy in the CGMs to increased accuracy, but this is not documented in the thesis.

As the derivation of the mapping is completely analytical it does not suffer from the unwieldiness that often characterizes a 'numerical' solution (such as the neural network solution). However, the stability and applicability of an analytical solution is not a priori guaranteed, and one should indeed expect to encounter some problems when applying the analytical solution to real measurements. This testing has not been carried out, but the author has briefly experimented with adding random noise to true measurements to estimate the stability of the mapping (not reported in the thesis). This revealed that for mild noise the mapping behaves nicely, but for medium to severe noise there is a significant stability issue.

One question that will appear time and again is what the optimal positions of the emitters and receivers are. In the present model they were allowed to be located anywhere in a 2D plane, and there was therefore a multi-dimensional infinite set of positions to choose

from. While there is no immediate answer to the question of an best set of positions (as this depends on the meaning of best) there is indeed a unique set of positions which can be regarded as being worst. This worst set of positions is, perhaps surprisingly, when emitters and receivers are located in the corners of a square, as this leads to a singularity in the mapping.

13.2.3 Modeling Reflection Maps

The reflection map modeling describes a setup where an emitter and a receiver is located some distance apart and facing in the same direction, and where the object is a ball of a given radius. By a ray-tracing-like method the reflected intensity of the ball in a given position is determined. The model includes the directional characteristics of the emitter and receiver, and the reflection characteristic of the ball. The model is constructed as a Fredholm integral equation of the first kind, and the model parameters are positions of emitter and receiver, spatial extension of the receiver, and radius of the ball.

The model of the reflection map was compared to the measured reflection map. Using angular difference between contour lines an exhaustive search in the parameter space yielded the optimal choice of parameters. The resulting set of parameters were quite close to the true values, and the modeled reflection map was in general similar to the measured reflection map. In particular, the measured map exhibits a characteristic non-symmetric structure which was recreated by the model. However, to achieve this it was necessary to (crudely) introduce the third dimension in the model.

13.3 Mathematics for Signal Processing (Part III)

The signal processing methods presented in the first part of the thesis are based on mathematical theory. In Part III some theory concerning transformations were presented in detail. The two subjects addressed are wavelet transformation of finite signals, and the Rudin-Shapiro transform. Though both subjects are indeed relevant in the design of signal processing algorithms for low-cost sensors, the presentations in Part III are mainly of mathematical nature without specific aim for applications, as this is covered in Part I.

13.3.1 Wavelet Transform

The wavelet transformation of finite signals is a matter which cannot be ignored when one wants to apply the WT. The wavelet theory is based on infinite signals, and some modification is necessary to handle finite signals. A number of methods have been reported in the literature, and a brief review of the most common methods were given in Chapter 9. None of these methods are ideal, but they do provide a variety of properties which are useful in different applications. However, the desire of the author to use a method capable of mapping low frequency noise in a well-defined and proper manner could not be fulfilled by any of the traditional methods. Therefore the more sophisticated moment

preserving wavelet transform suggested by Cohen, Daubechies, and Vial have been investigated. This transform has be ability to map polynomials into polynomials in the low pass part and to the zero sequences in the high pass part.

The introduction of this transform takes up quite a few pages as the construction is presented from scratch. This is done partly because the author felt the need for understanding the construction in detail, partly because a rather thorough presentation is necessary in order to implement the construction in MATLAB. This code is printed in the appendices. The software implementation allowed the author to experiment with various filters, and it turned out that the construction has an inherent stability problem. One of the few filters which do result in a relatively stable transform is the ones (Symlets) used in the original paper. Unfortunately, the author has not been able to provide a useful suggestion for handling this instability.

13.3.2 Rudin-Shapiro Transform

Compared to the wavelet transform, on which their exists a huge number of publications, the Rudin-Shapiro transform is almost unknown. The primary reason for using the RST in this thesis is its spread spectrum property. The concept of spread spectrum is well-established in the signal processing society, but apparently the idea of systematizing the construction of pseudo-random binary sequences by means of a transform with a fast implementation is not wide-spread. Therefore, the RST is presented in some detail in this thesis. A number of useful properties are demonstrated, including some which are important in applications.

The use of spread spectrum sequences in active sensors is most likely not a new idea, although the author has not been able to find any references to existing sensors employing this technique. It should be noted that SS sequences can be generated in a number of different ways. The transform approach suggested in this thesis is highly structured and allows for easy adjustment of various parameters depending on the given sensor implementation. This freedom is not that easily achieved in solutions where more ad hoc-like methods are used. This is not to say that the RST is useful in all spread spectrum systems. I many cases properties not provided by the RST are important, such as a certain structure of the cross-correlations and the possibility for any SS sequence length.

13.4 Future Work

Throughout the thesis a number of subjects have been brought up. Not all of these have been investigated or discussed to an extent which allows the author to consider them exhausted. This section provides a list of these subjects with a brief comment on what further work could be done. A somewhat more detailed discussion of possible future work on spatial position sensors is given in Section 6.4. Note that the subjects listed here only includes specific problems encountered in the thesis.

Validation on several channels jointly The validation methods presented in this thesis

are focused on validating each channel separately. However, since in some systems a number of 'simultaneous' channels need to validated, it is obvious to consider whether it would an advantages to validate the channels jointly. It is expected, however, that the benefits of doing this are minimal.

Better statistical model for validation The statistical model presented in Section 4.9 can be improve in a number of ways. For instance, the current model assumes y_0 to be a deterministic signal, and that σ is fairly accurately known.

A more thorough investigation of neural networks The neural network approach presented in Chapter 6 was not investigated sufficiently for any conclusions to be drawn. While the method does present some interesting prospects it is still to be determined how to parameterized such an approach, how to keep complexity sufficiently low, whether a radial basis function network is the way to go. Many other questions remain, too.

Modeling of 3D sub-manifold for denoising It was suggested that denoising of CGM data could be accomplished by projection onto a 3D sub-manifold. It (probably) requires a significant effort to determine the feasibility of this suggestion. That is, whether the method would work, and whether it can be implemented within reasonable time and programming limits.

Geometrical modeling with a more accurate reflection model The geometrical modeling in Chapter 7 provided a mapping from CGMs to spatial position. However, the model was based on a series of assumptions that are arguably not sufficiently realistic. The development of a really useful model is therefore still to be accomplished. In particular, the reflection model presented in Chapter 8 must somehow by included.

Better modeling of the 'emitter, receiver, reflecting object' setup The reflection map model was not too bad at generating a realistic reflection map. However, there were indications (such as the result of the inverse problem considerations) that the model might not be sufficiently accurate for predicting the reflected intensity in the entire space in front of the emitter and receiver. One obvious extension of the model is to include the third dimension.

Making the moment preserving wavelet transform stable The thesis has not provided any really useful suggestion for restoring the numerical stability in the moment preserving edge filter construction. It was mentioned that including more interior scaling functions might be a way to solve the problem, but this requires an update of the construction, and this has not been accomplished in this thesis.

Proving the near-flatness of the Rudin-Shapiro polynomials on $(0; 1/2)$ Despite several attempts the author has not been able to verify the conjecture that Rudin-Shapiro polynomials are near-flat on $(0; 1/2)$. One might think this indicates that the conjecture is false, but it is common in the field of flat polynomials that even simple conjectures can be quite difficult to prove.

Proving a number of conjectures Throughout the thesis a few propositions of various kind has been conjectured. These are still open for validation (or rejection).

Appendices

Basic Properties of the Wavelet Transform

A

The construction of orthonormal wavelet bases and of pairs of dual, biorthogonal wavelet bases for $L^2(\mathbb{R})$ is now well understood. For the construction of orthonormal bases of compactly supported wavelets for $L^2(\mathbb{R})$, in particular, one starts with a trigonometric polynomial $m_0(\xi) = \sum_n c_n e^{-in\xi}$ satisfying $m_0(0) = 1$ and $|m_0(\xi)|^2 + |m_0(\xi + \pi)|^2 = 1$ as well as some mild technical conditions. A sufficient but not necessary condition, always satisfied in practice, is $|m_0(\xi)| \neq 0$ for all $|\xi| \leq \pi/2$ (see for instance Mallat [55]). The corresponding scaling function ϕ and wavelet ψ is then defined by

$$\hat{\phi}(\xi) = \frac{1}{\sqrt{2\pi}} \prod_{j=1}^{\infty} m_0(2^{-j}\xi) \qquad \text{and} \qquad \hat{\psi}(\xi) = e^{-i\xi/2}\overline{m_0(\xi/2 + \pi)}\hat{\phi}(\xi/2). \quad \text{(A.1)}$$

Here $\hat{}$ denote the Fourier transform normalized as $\hat{f}(\xi) = (2\pi)^{-1/2} \int f(t) e^{-i\xi t} dt$. The functions

$$\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k), \quad j, k \in \mathbb{Z},$$

constitutes an orthonormal basis for $L^2(\mathbb{R})$. For fixed j, the

$$\phi_{j,k}(t) = 2^{-j/2}\phi(2^{-j}t - k), \quad k \in \mathbb{Z}$$

are an orthonormal basis for a subspace $V_j \subset L^2(\mathbb{R})$, and the spaces V_j constitute a multiresolution analysis, meaning that

$$\cdots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \cdots,$$

with

$$\bigcap_{j \in \mathbb{Z}} V_j = 0, \qquad \overline{\bigcup_{j \in \mathbb{Z}} V_j} = L^2(\mathbb{R})$$

and

$$\text{Proj}_{V_{j-1}} f = \text{Proj}_{V_j} f + \sum_{k \in \mathbb{Z}} \langle f, \psi_{j,k} \rangle \psi_{j,k}.$$

A consequence of the first equation in (A.1) is $\hat{\phi}(\xi) = m_0(\xi/2)\hat{\phi}(\xi/2)$, and the inverse Fourier transform of this is

$$\phi(t) = \sqrt{2} \sum_n h_n \phi(2t - n).$$

The filter taps **h** (and the corresponding **g** originating from the second equation in (A.1)) is used for the time domain discrete wavelet transform of the vector **x** as

$$y_k^0 = \sum_n h_{2k-n} x_n = \sum_n h_n x_{2k-n}, \tag{A.2}$$

$$y_k^1 = \sum_n g_{2k-n} x_n = \sum_n g_n x_{2k-n}. \tag{A.3}$$

Extra Lemmas, Expressions, and Figures

B

Lemma B.1
Define the $N \times N$ matrix $\mathbf{\Gamma}$ as

$$\gamma_{m,n} = \begin{cases} 1 & m = n, m \leq 1 \\ \dfrac{\gamma_{m-1,n-1} - (m-1)\gamma_{m-1,n}}{m} & 1 \leq n \leq m-1, m \geq 2 \\ 0 & \textit{otherwise}, \end{cases} \tag{B.1}$$

for $m, n = 0, \ldots, N-1$. Then $\mathbf{\Gamma}$ is invertible and

$$y^k = \sum_{n=0}^{k} \binom{y}{n} \tilde{\gamma}_{k,n}, \quad k = 0, \ldots, N-1 \tag{B.2}$$

where $\tilde{\mathbf{\Gamma}} = \mathbf{\Gamma}^{-1}$.

Proof
First note that

$$\sum_{n=1}^{k} \gamma_{k,n} y^n = \binom{y}{k}, \quad k \geq 1. \tag{B.3}$$

This is seen by induction. Assume (B.3) holds for k. Applying (B.1) to (B.3) yields

$$\begin{aligned} \sum_{n=1}^{k+1} \gamma_{k+1,n} y^n &= \frac{1}{k+1} \sum_{n=1}^{k+1} (\gamma_{k,n-1} - k\gamma_{k,n}) y^n \\ &= \frac{1}{k+1} \Big(\sum_{n=1}^{k} \gamma_{k,n} y^{n+1} - \sum_{n=1}^{k} k\gamma_{k,n} y^n \Big) = \frac{y-k}{k+1} \binom{y}{k} = \binom{y}{k+1}. \end{aligned}$$

Together with $\binom{y}{1} = y$ this demonstrates that (B.3) also holds for $k+1$. Because $\mathbf{\Gamma}$ is lower triangular and $\gamma_{m,m} = m!^{-1}$, the matrix is invertible. The equation (B.2) follows immediately from (B.3) and the matrix inversion. □

Lemma B.2
The polynomial $f(x) \geq 0$ *iff* $f(x) = h^2(x) + g^2(x)$*, where* h *and* g *are polynomials.*

Proof
When $f(x) \geq 0$ the solutions to $f(x) = 0$ are N real (and therefore double) roots r_n, and M complex roots $a_m \pm i b_m$. Hence

$$\begin{aligned} f(x) &= k^2 \prod_{n=1}^{N} (x - r_n)^2 \prod_{m=1}^{M} [x - (a_m + i b_m)][x - (a_m - i b_m)] \\ &= k^2 \prod_{n=1}^{N} (x - r_n)^2 \prod_{m=1}^{M} (x - a_m)^2 + b_m^2 \\ &= k^2 (p_a^2(x) + p_b^2(x)) \prod_{n=1}^{N} (x - r_n)^2 \qquad (*) \\ &= \left[k p_a(x) \prod_{n=1}^{N} (x - r_n) \right]^2 + \left[k p_b(x) \prod_{n=1}^{N} (x - r_n) \right]^2, \end{aligned}$$

where $(*)$ follows from

$$(a^2 + b^2)(c^2 + d^2) = (ac + bd)^2 + (ad - bc)^2.$$

The other way is trivial. □

Lemma B.3
The function

$$p(x, y) = x^2 y^2 (x^2 + y^2 - 1) + 1 \tag{B.4}$$

is bounded below by $26/27$ *and there does not exist polynomials* $q_n(x, y)$*,* $n = 1, \ldots, N$ *such that*

$$p(x, y) = \sum_{n=1}^{N} q_n^2(x, y). \tag{B.5}$$

Proof
Solving

$$\frac{\partial}{\partial x} p(x, y) = \frac{\partial}{\partial y} p(x, y) = 0 \tag{B.6}$$

leads to $y^4 - x^4 = 0$, and therefore a necessary condition for $p(x, y)$ to be minimal is $x = y$. Since the solutions to

$$\frac{\partial}{\partial z} z^2 (2z - 1) + 1 = 0$$

is 0 and $1/3$ it follows that $p(x, y)$ is bounded below by $p(1/\sqrt{3}, 1/\sqrt{3}) = 26/27$.

The fact that there does not exist polynomial fulfilling (B.5) is proven by contradiction. Therefore, assume that the polynomials q_n do exist. Since $p(x, 0) = p(0, y) = 1$ we get that $q_n(x, 0)$ and $q_n(0, y)$ must be constant for $n = 1, \ldots, N$. Therefore each q_n can be written

$$q_n(x, y) = a_n + xyh_n(x, y),$$

where a_n is a constant and h_n is of degree at most 1 (there may be x and y terms in h_n, but no xy term, since this would violate the degree of $p(x, y)$). By comparing terms of the same degree we find

$$\sum_{n=1}^{N} a_n^2 = 1, \qquad 2xy \sum_{n=1}^{N} a_n h_n(x, y) = 0$$

and therefore

$$x^2y^2(x^2 + y^2 - 1) = x^2y^2 \sum_{n=1}^{N} h_n^2(x, y).$$

Then

$$x^2 + y^2 - 1 = \sum_{n=1}^{N} h_n^2(x, y)$$

which is a contradiction. □

The proof was found (with slightly fewer details) in [7, p. 190-191]

The extended proof of Lemma 7.6 is given here. The expressions have been generated in Maple V R6 and rewritten to shorter form by hand.

Proof of Lemma 7.6
Expanding the three spheroids (7.6), (7.7), and (7.8) yields

$$\begin{aligned}\big((d_1 - d_2)^2 - w_1^2r^2\big)x^2 + 2\big((y - 1 + d_1^2 - d_2^2)(d_2 - d_1) + (d_1 + d_2)w_1^2r^2\big)x \\ + w_1^4r^4 + \big(2y - 2 - y^2 - z^2 - 2(d_1^2 + d_2^2)\big)w_1^2r^2 \\ + 2(d_1^2 - d_2^2)(y - 1) + (y - 1)^2 + (d_1^2 - d_2^2)^2 = 0\,, \quad \text{(B.7)}\end{aligned}$$

$$\begin{aligned}(d_2^2 - w_2^2r^2)x^2 + 2d_2(y + w_2^2r^2 - d_2^2 - 1)x + w_2^4r^4 + \\ (2y - 2d_2^2 - z^2 - y^2 - 2)w_2^2r^2 + (d_2^2 + 2 - 2y)d_2^2 + (y - 1)^2 = 0\,, \quad \text{(B.8)}\end{aligned}$$

$$(w_3^2r^2 - d_1^2)x^2 + 2d_1(d_1^2 - w_3^2r^2)x - d_1^4 + (z^2 + y^2 + 2d_1^2 - w_3^2r^2)w_3^2r^2 = 0\,. \quad \text{(B.9)}$$

Modify (B.8) by the ratio of the z^2 coefficients in (B.9) and (B.8), and subtract (B.8). Solving this yields

$$x_1 = \frac{-w_3^2w_2r^2 + (y - 1 - d_2^2 + w_2^2r^2)w_3 + w_2d_1^2}{w_2d_1 - w_3d_2} \qquad \text{(B.10)}$$

$$x_2 = -\frac{w_3^2 w_2 r^2 + (y - 1 - d_2^2 + w_2^2 r^2) w_3 - w_2 d_1^2}{w_2 d_1 + w_3 d_2} \tag{B.11}$$

Now, insert x_1 in (B.7) and (B.8), and modify (B.7) by the ratio of the z^2 coefficients, and subtract (B.8). Solving this yields

$$y_{11} = 1 + d_2^2 - d_1 d_2 + (w_3 - w_1) w_2 r^2 + \frac{(w_1 - w_2) w_3 d_2 r^2}{d_1}$$
$$y_{12} = 1 + d_2^2 - d_1 d_2 + (w_1 + w_3) w_2 r^2 - \frac{(w_1 + w_2) w_3 d_2 r^2}{d_1}$$

and

$$x_{11} = \frac{(w_2 - w_1) w_3 r^2}{d_1} + d_1$$
$$x_{12} = \frac{(w_2 + w_1) w_3 r^2}{d_1} + d_1$$

The same procedure applied to x_2 yields

$$y_{21} = d_1 d_2 - 1 - d_2^2 + (w_3 + w_1) w_2 r^2 + \frac{(w_2 - w_1) d_2 w_3 r^2}{d_1}$$
$$y_{22} = d_1 d_2 - 1 - d_2^2 + (w_3 - w_1) w_2 r^2 + \frac{(w_2 + w_1) d_2 w_3 r^2}{d_1}$$

$$x_{21} = -\frac{(w_2 + w_1) w_3 r^2}{d_1} + d_1$$
$$x_{22} = -\frac{(w_2 - w_1) w_3 r^2}{d_1} + d_1$$

Inserting the four solution sets in (B.8) yields eight z's of which only four are positive (as required). Inserting a $\mathbf{w}$ corresponding to an arbitrary $(x, y, z) \in \mathbb{R}^2 \otimes \mathbb{R}^+$ in each of the four solution sets reveals that only x_{11}, y_{11} with the corresponding z belongs to $\mathbb{R}^{3+}$. □

The three expressions for p_1, p_2, and p_3 in (7.17) are

$$
\begin{aligned}
p_1 &= d_1 w_3 \big[(-2d_2^2 w_1 w_3^2 w_2 + w_1^2 w_3^2 d_2^2 + d_2^2 w_3^2 w_2^2 - 2w_3^2 w_2 w_1 \\
&\quad + d_1^2 w_3^2 w_2^2 + w_2^2 w_1^2 d_1^2 + 2d_2 w_1 w_3^2 d_1 w_2 - 2w_2 d_2 w_1^2 d_1 w_3 + 2d_2 w_3 w_2^2 w_1 d_1 \\
&\quad + w_3^2 w_1^2 + w_3^2 w_2^2 - 2w_1 d_1^2 w_2^2 w_3 - 2d_1 d_2 w_3^2 w_2^2)/((w_1 - w_2)^2 w_3^2)\big]^{1/2} \\
&\quad \times (w_1 w_3^2 + w_1^3 - w_3^2 w_2 - w_2^3 - 3w_1^2 w_2 + 3w_1 w_2^2 - 2w_3 w_1^2 - 2w_2^2 w_3 + 4w_3 w_2 w_1) \\
&= d_1 (w_1 - w_3 - w_2)^2 \sqrt{(d_1 w_2 w_{13} - d_2 w_3 w_{12})^2 + w_3^2 w_{12}^2}\,, \\
p_2 &= d_1^2 w_1^3 w_2 - w_3^2 w_2^2 - 2w_1^2 w_3^2 d_2^2 - 2d_2^2 w_3^2 w_2^2 + 2w_3^2 w_2 w_1 - 2d_1^2 w_3^2 w_2^2 \\
&\quad - 4d_2 w_1 w_3^2 d_1 w_2 + 3w_2 d_2 w_1^2 d_1 w_3 - 3d_2 w_3 w_2^2 w_1 d_1 - 2w_2^2 w_1^2 d_1^2 - w_3^2 w_1^2 \\
&\quad + 4d_2^2 w_1 w_3^2 w_2 + 5w_1 d_1^2 w_2^2 w_3 + 2d_1 d_2 w_3^2 w_2^2 + d_1^2 w_1^3 w_3 + d_1^2 w_1 w_2^3 \\
&\quad + d_1^2 w_1 w_3^3 - w_2^2 w_3^2 d_2^4 - 2d_1^2 w_2^3 w_3 - d_2^4 w_1^2 w_3^2 - 2w_3^3 d_1^2 w_2 - 2d_1^2 w_3^2 w_1^2 \\
&\quad - d_1^4 w_3^2 w_2^2 - w_2^2 w_1^2 d_1^4 + w_2^3 w_3 d_2 d_1 + w_2 w_3^3 d_2 d_1 - 2w_2^2 w_3^2 d_2^2 d_1^2 \\
&\quad + 2w_2 w_3^2 d_2^4 w_1 + 2w_2^2 w_3^2 d_2^3 d_1 + 2d_1 d_2^3 w_1^2 w_3^2 + d_1^2 d_2^2 w_1 w_3^3 \\
&\quad + d_1^2 d_2^2 w_1^3 w_3 - d_1^3 d_2 w_1 w_2^3 - d_1^3 d_2 w_1^3 w_2 + d_1 w_2^3 d_2^3 w_3 + d_1^3 w_2^3 d_2 w_3 \\
&\quad + d_1^2 w_2^3 d_2^2 w_1 - 2d_1^2 w_2^3 d_2^2 w_3 - d_2 w_1 w_3^3 d_1 - d_2 w_1^3 w_3 d_1 + 2w_3^2 d_1^3 w_2^2 d_2 \\
&\quad - 4w_3 d_1^2 w_2 w_1^2 + d_1 w_3^3 w_2 d_2^3 + d_1^3 w_3^3 w_2 d_2 - d_1 w_3^3 d_2^3 w_1 - 2d_1^2 w_3^3 d_2^2 w_2 \\
&\quad + d_1^2 d_2^2 w_1^3 w_2 - 4w_2 w_3^2 d_2^3 d_1 w_1 + 5w_2^2 w_3 d_2^2 d_1^2 w_1 - 3w_2^2 w_3 d_2^3 d_1 w_1 \\
&\quad + 5d_1^2 d_2^2 w_1 w_3^2 w_2 - 4d_1^2 d_2^2 w_1^2 w_2 w_3 - 4d_1^3 d_2 w_1 w_2^2 w_3 - 3d_1^3 d_2 w_1 w_2 w_3^2 \\
&\quad + 3d_2^3 w_1^2 w_3 d_1 w_2 + 3w_2 d_1^3 d_2 w_3 w_1^2 - d_2^3 w_1^3 w_3 d_1 - 2w_1^2 w_3^2 d_2^2 d_1^2 \\
&\quad + 5w_3^2 w_2 w_1 d_1^2 - 2w_2^2 w_1^2 d_1^2 d_2^2 + 2w_2^2 w_1^2 d_1^3 d_2 + 2w_3^2 w_1^2 d_1 d_2 + 2w_1 d_1^4 w_2^2 w_3 \\
&= d_1^2 (d_2^2 + 1)\big((w_1 - 2w_3) w_2^3 + (w_1 - 2w_2) w_3^3 + (w_3 + w_2) w_1^3\big) \\
&\quad - d_1 d_2 (w_1 - w_3 - w_2)^2 \big(w_2 w_{13} d_1^2 + w_3 w_{12} (d_2^2 + 1)\big) \\
&\quad - (d_2^2 w_{12} w_3 + w_2 w_{13} d_1^2)^2 - w_{12}^2 w_3^2 (1 + 2d_2^2) \\
&\quad - d_1^2 \big(w_1 (w_3 + w_2) - 2w_3 w_2\big)\big(2w_1 (w_3 + w_2) - w_3 w_2\big) \\
&\quad - d_1^2 d_2^2 w_1 \big(w_3 w_2 (2w_1 - 3w_2 - 3w_3) + 2w_1 (w_3^2 + w_2^2)\big)\,, \\
p_3 &= -2d_2^2 w_1 w_3^2 w_2 + w_1^2 w_3^2 d_2^2 + d_2^2 w_3^2 w_2^2 - 2w_3^2 w_2 w_1 + d_1^2 w_3^2 w_2^2 + w_2^2 w_1^2 d_1^2 \\
&\quad + 2d_2 w_1 w_3^2 d_1 w_2 - 2w_2 d_2 w_1^2 d_1 w_3 + 2d_2 w_3 w_2^2 w_1 d_1 + w_3^2 w_1^2 + w_3^2 w_2^2 \\
&\quad - 2w_1 d_1^2 w_2^2 w_3 - 2d_1 d_2 w_3^2 w_2^2 \\
&= \big(w_{12}^2 + d_{12}^2 w_2^2 + d_2 w_1 (w_1 d_2 + 2w_2 d_{12})\big) w_3^2 - (d_2 w_{12} + d_1 w_2) 2d_1 w_1 w_2 w_3 \\
&\quad + (d_1 w_1 w_2)^2\,.
\end{aligned}
$$

The expanded versions of p_1, p_2, and p_3 have been obtained in Maple, while the more compact versions have been derived manually from the expanded versions.

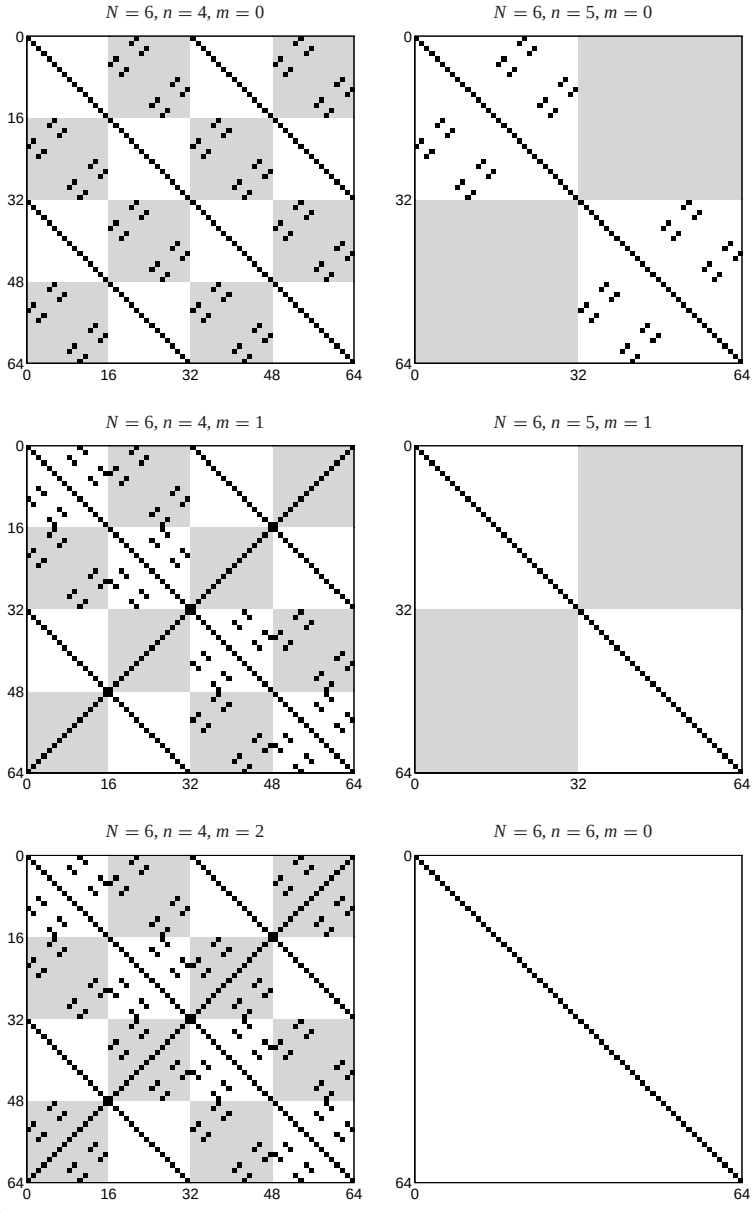


Figure B.1: Result of applying polynomial denoising in a block diagonal structure to an entire RST. The matrix has size 2^N, the number of signal parts is 2^n, the polynomial has degree m.

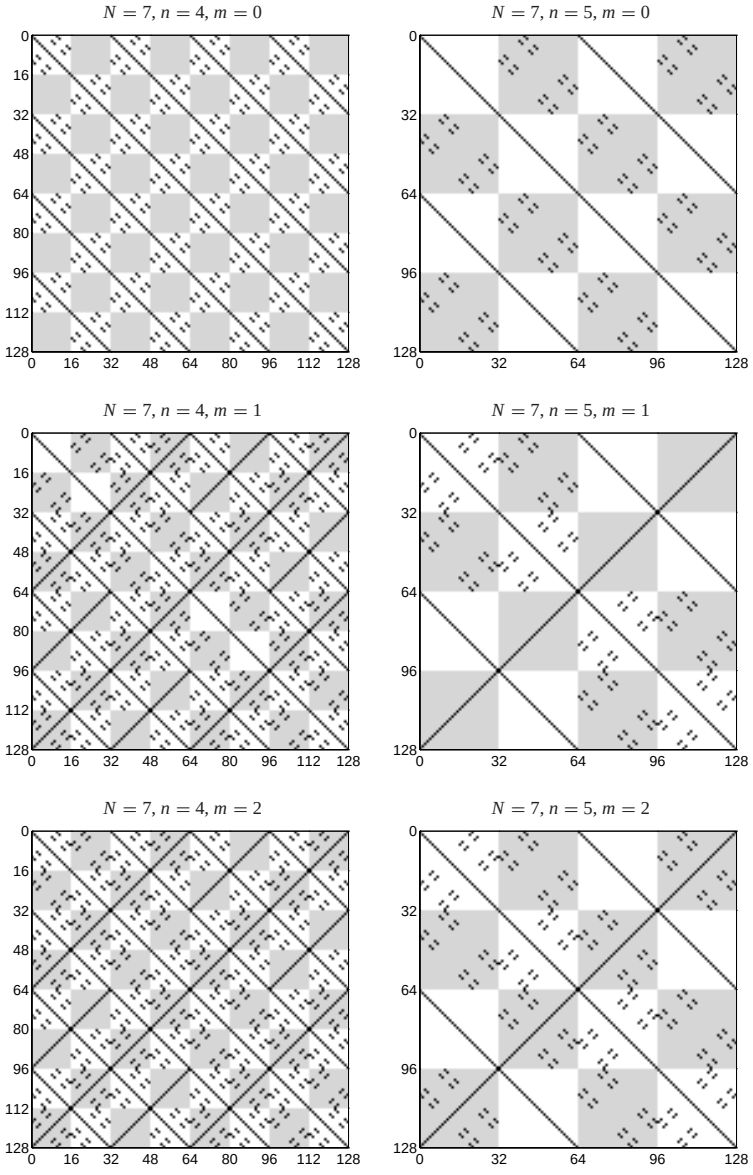


Figure B.2: Result of applying polynomial denoising in a block diagonal structure to an entire RST. The matrix has size 2^N, the number of signal parts is 2^n, the polynomial has degree m.

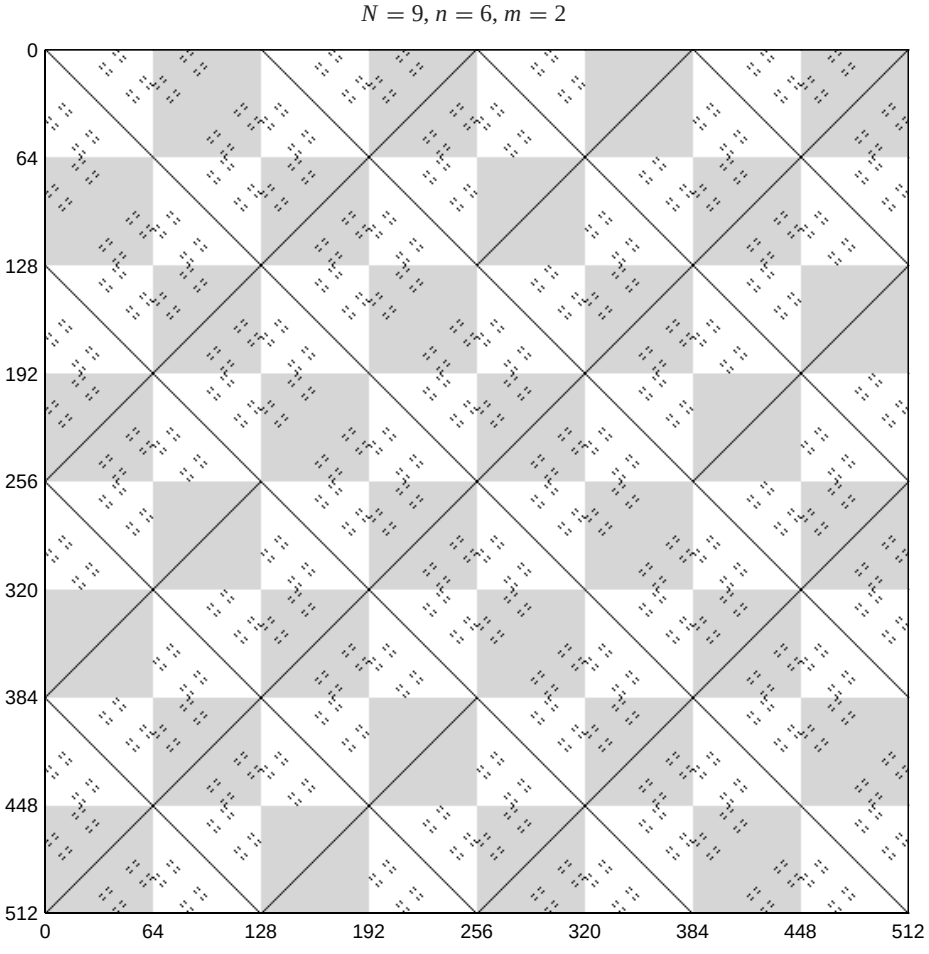


Figure B.3: Result of applying polynomial denoising in a block diagonal structure to an entire RST. The matrix has size 2^N, the number of signal parts is 2^n, the polynomial has degree m.

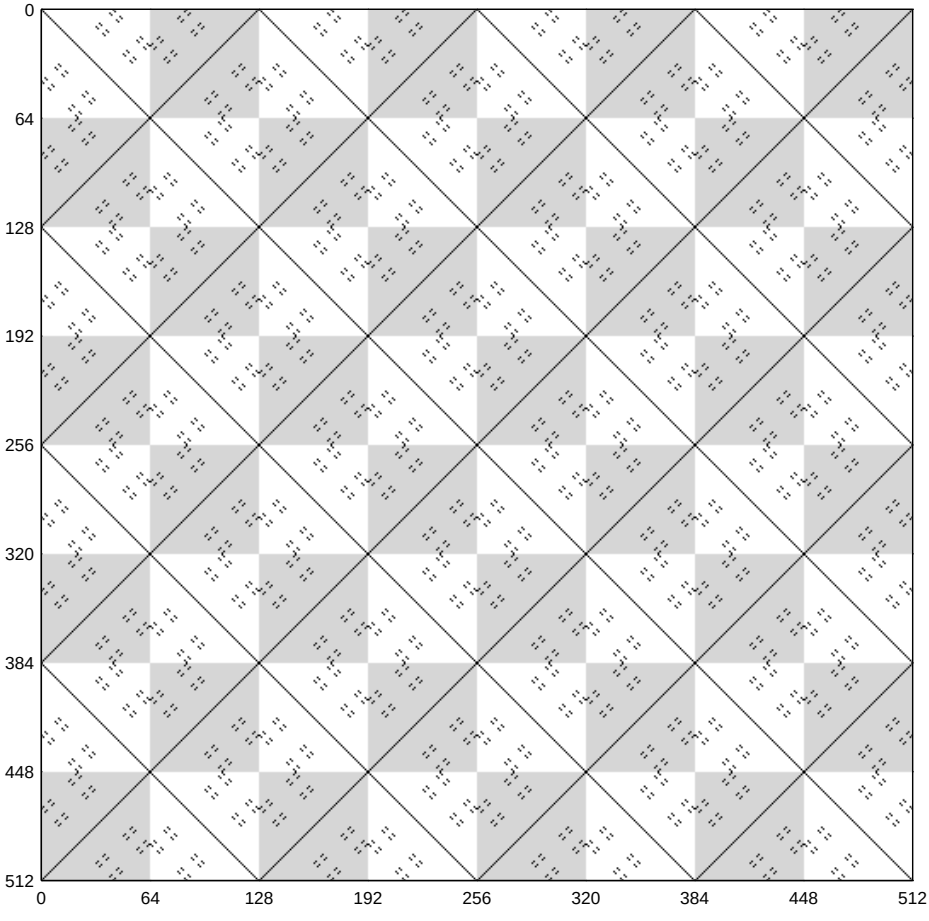


Figure B.4: Result of applying the WT in a block diagonal structure to an entire RST. The matrix has size 2^9 and the number of signal parts/WT blocks is 2^6.

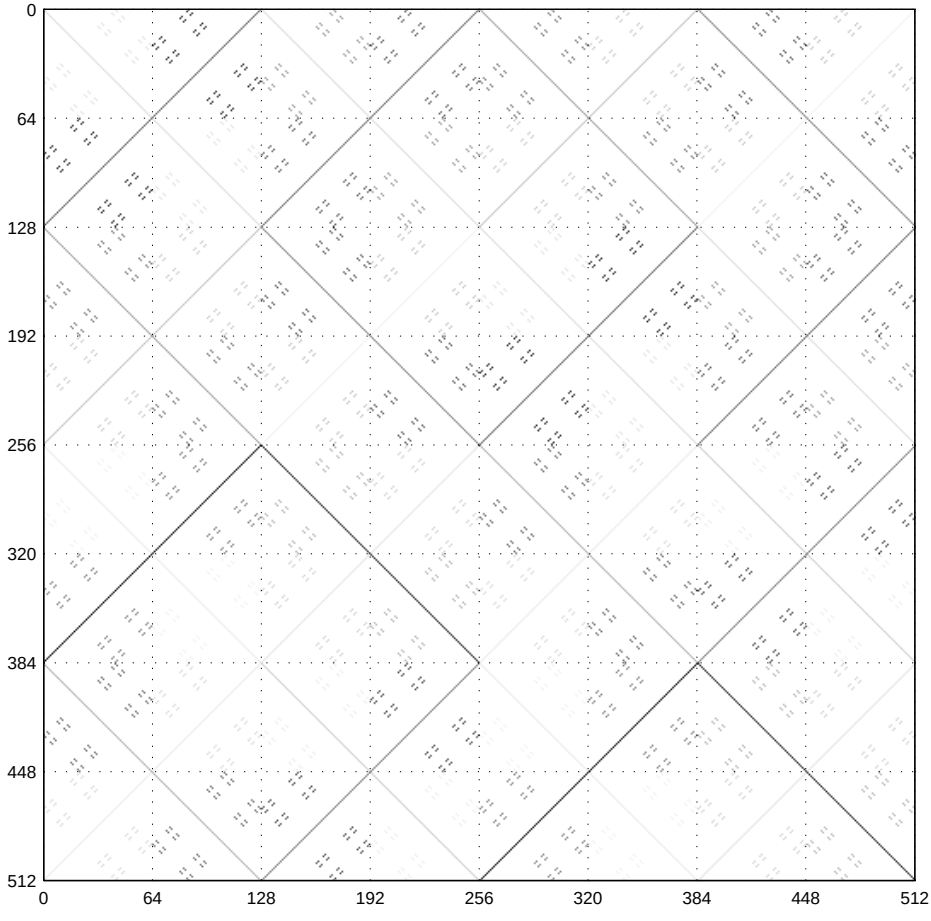


Figure B.5: Result of applying the WT in a block diagonal structure to an entire RST. The matrix has size 2^9 and the number of signal parts/WT blocks is 2^6.

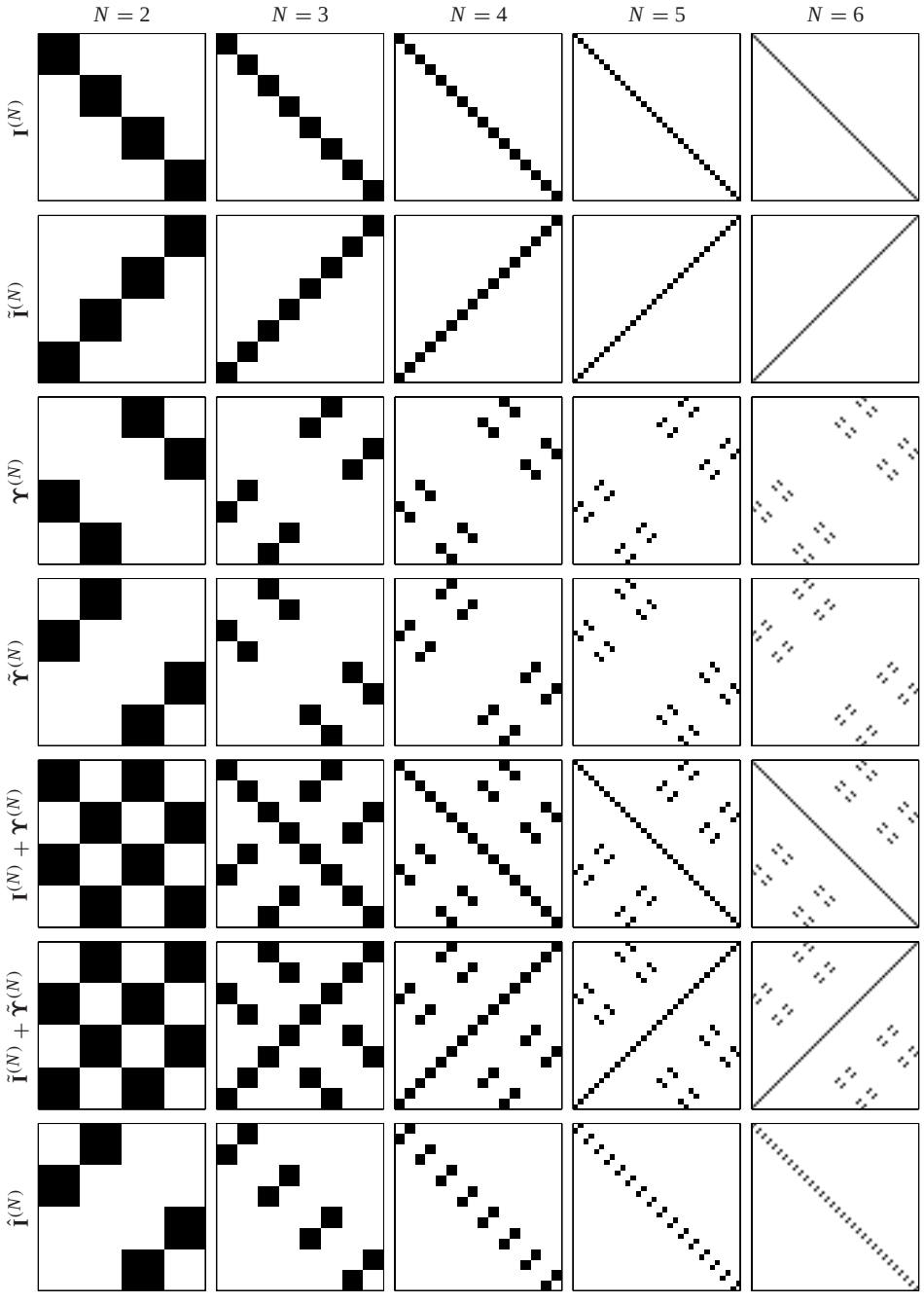


Figure B.6: Each column shows the set of the seven different structures of the $\boldsymbol{\Upsilon}$ and $\mathbf{I}$ related matrices for a particular N.

Moment Preserving Edge Filters in Matlab

C

Function 1 Generation of All Edge Filter Coefficients

```
function [Ledge,Redge,Aleft,Aright] = edgecoef(h);

% EDGECOEF    Calculate edge coefs for polynomial recontruction
%
% Syntax: [Ledge, Redge, AL, AR] = EDGECOEF(H)
%
%    The low and high pass left and right edge coefficients
%    used for wavelet transform on the interval where polynomial
%    regeneration is desired. AL and AR are the preconditioning
%    matrices.
%
% Reference: Cohen, Daubechies, Vial: Wavelets on the interval.
%            Appl. and Comp. Harm. Anal. vol 1, no. 1,
%            december 1993.

[Hleft,Gleft,Aleft] = EdgeFilterCoefs(h);
Hleft = round(Hleft*1e13)/1e13;
Gleft = round(Gleft*1e13)/1e13;
Aleft = fliplr(flipud(Aleft));

[Hright,Gright,Aright] = EdgeFilterCoefs(fliplr(h));
Hright = round(flipud(fliplr(Hright))*1e13)/1e13;
Gright = round(flipud(fliplr(Gright))*1e13)/1e13;

Ledge = [Hleft; Gleft];
Redge = [Hright; Gright];
```

Function 2 Edge Filter Coefficients

```
function [Hedge, Gedge, Aedge] = EdgeFilterCoefs(h)

N = length(h)/2;

[A,B,E,Et] = EdgeABE(h);
```

```
% Calculate the H_{m,s}
Hedge = zeros(N,2*N);
for m = 0:N-1
  for s = N:N+2*m
    Acc = 0;
    for k = N-1-m:N-1
      Acc = Acc + E(N-1-m+1,k+1)*B(k+1,s+1-N);
    end
    Hedge(m+1,s+1) = Acc/sqrt(2);
  end
end

% Calculate the h_{m,s}
for m = 0:N-1
  for s = 0:N-1
    Acc = 0;
    for n = 0:N-1
      for k = n:N-1
        Acc = Acc + Et(n+1,N-1-s+1)*E(N-1-m+1,k+1)*A(k+1,n+1);
      end
    end
    Hedge(m+1,s+1) = Acc/sqrt(2);
  end
end

% Calculate the Gedge
Gedge = eye(3*N-1) - Hedge'*Hedge;
Gedge = Gedge(1:N,:);

% Gaussian elimination from the bottom up
for k = N-1:-1:1
  for n = k:-1:1
    r = (Gedge(n,2*k+N)/D(k+1,2*k+N));
    Gedge(n,:) = Gedge(n,:) - Gedge(k+1,:)*r;
  end
end

% Orthonormalization
for k = 1:N
  for n = 1:k-1
    Gedge(k,:) = Gedge(k,:) - Gedge(n,:)*Gedge(k,:)'*Gedge(n,:);
  end
  Gedge(k,:) = Gedge(k,:) / norm(Gedge(k,:));
end

% Calculate the preconditioning matrix
V = zeros(N);
for m=1:N
  for n=1:m
    V(n,m) = nchoosek(m-1,n-1);
```

```
  end
end
Aedge =  inv(V) * Et;
```

Function 3 Sampled Continuous Edge Functions

```
function [SL,SR,WL,WR,T] = edgefunc(h,j);

% EDGEFUNC        Calculate the continuous edge functions
%
% Syntax: [SL,SR,WL,WR,T] = EDGEFUNC(H [,J])
%
%   Calculate the continuous edge functions in 2^J points
%   per unit. S is scaling function, and W wavelets
%   at repsectively the left and right end.
%   T is the proper time scale.
%
%   This function requires the Uvi_Wave Toolbox to be installed.

if nargin == 1 j = 6; end

N = length(h)/2;

for i=1:length(h)
  g(i) = -(-1)^i*h(2*N-i+1);
end

% Left edge scaling functions
[A,B,E,Et] = EdgeABE(h);
s = wavelet(h,g,j)*sqrt(2);
s = [s zeros(1, (2*N-1)*2^j-length(s))];

BN = zeros(N,2*N-1);
for k=1:N
  for n=k:2*N-1
    BN(k,n) = nchoosek(n-1,k-1);
  end
end

M = zeros(2*N-1,(2*N-1)*2^j);
for k=1:2*N-1
  M(2*N-k,1:2^j*k) = s(end-k*2^j+1:end);
end
SL = flipud(E * BN * M);

% Left edge wavelets
[Ledge,Redge] = edgecoef(h);
Hleft = Ledge(1:N,:);
Gleft = Ledge(N+1:2*N,:);
```

```
Hright = Redge(1:N,:);
Gright = Redge(N+1:2*N,:);

WL = zeros(N,(2*N-1)*2^j);
for k=1:N
  WL(k,1:(2*N-1)*2^(j-1)) = Gleft(k,1:N) * SL(:,1:2:end);
  for m=N:N+2*(k-1)
    WL(k,1+(m-N+1)*2^(j-1):(N+m)*2^(j-1)) = ...
         WL(k,1+(m-N+1)*2^(j-1):(N+m)*2^(j-1)) ...
         + Gleft(k,m+1)*s(1:2:end);
  end
end
WL = WL * sqrt(2);

% Right edge scaling functions
[A,B,E,Et] = EdgeABE(fliplr(h));
s = wavelet(fliplr(h),fliplr(g),j)*sqrt(2);
s = [s zeros(1, (2*N-1)*2^j-length(s))];

M = zeros(2*N-1,(2*N-1)*2^j);
for k=1:2*N-1
  M(2*N-k,1:2^j*k) = s(end-k*2^j+1:end);
end
SR = fliplr(E * BN * M);

% Right edge wavelets
WR = zeros(N,(2*N-1)*2^j);
for k=1:N
  WR(k,(2*N-1)*2^(j-1)+1:end) = ...
      Gright(k,end-N+1:end) * SR(:,1:2:end);
  for m=N:N+2*(N-k)
    St = 1+(m-N+2*k-2)*2^(j-1);
    En = (N+m-1+2*k-2)*2^(j-1);
    WR(k,St:En) = WR(k,St:En) ...
        + Gright(k,m-N+1+2*k-2)*fliplr(s(1:2:end));
  end
end
WR = WR * sqrt(2);

T = linspace(0,2*N-1,2^j*(2*N-1));
```

Function 4 Auxilary Edge Filter Matrices

```
function [A,B,E,Et] = EdgeABE(h);

N = length(h)/2;

% Calculate the gamma matrix
gamma = zeros(N);
```

```
gamma(1,1) = 1;
gamma(2,2) = 1;
for n = 3:N
  gamma(n,2:n) = (gamma(n-1,1:n-1) - (n-2)*gamma(n-1,2:n))/(n-1);
end
gammat = inv(gamma);

bin = zeros(N);
for m = 0:N-1
  for n = 0:m
    bin(m+1,n+1) = nchoosek(m,n);
  end
end
bin2 = bin.*fliplr(vander((N-1)*ones(1,N)));

% Calculate the alpha matrix
A = zeros(N);
hpow = h*fliplr(vander([-N+1:N]));
for k = 0:N-1
  for n = 0:k
    Acc = 0;
    for q = n:k
      for r = n:q
        Acc = Acc + 2^(-q)*gamma(k+1,q+1)*gammat(r+1,n+1)...
                  *nchoosek(q,r)*(bin2(q-r+1,1:q-r+1)...
                  *(hpow(:,q-r+1:-1:1))');
      end
    end
    A(k+1,n+1) = Acc/sqrt(2);
  end
end

% Calculate the beta matrix
B = zeros(N,3*N-2);
for k=0:N-1
  for n=N:3*N-2
    b = 0;
    for m=0:k
      c = 0;
      for s=-N+1:N-1
        if 2*s+n > -N & 2*s+n < N+1
          c = c + (s+N-1)^m*h(2*s+n+N);
        end
      end
      b = b + c*gamma(k+1,m+1);
    end
    B(k+1,n+1) = b*sqrt(2);
  end
end
B = B(:,N+1:end);
```

```
% Calculate the eta matrix
Eta = zeros(N);
for s = 0:N-1
  for k = 0:s
    Acc = 0;
    for m = 0:k-1
      for n = 0:s
        Acc = Acc + A(k+1,m+1)*A(s+1,n+1)*Eta(m+1,n+1);
      end
    end
    for n = 0:s-1
      Acc = Acc + 1/2^k*A(s+1,n+1)*Eta(k+1,n+1);
    end
    for m = N:3*N-2-2*min(k,s)
      Acc = Acc + B(k+1,m+1-N)*B(s+1,m+1-N);
    end
    Eta(k+1,s+1) = Acc/(2-2^(-k-s));
    Eta(s+1,k+1) = Eta(k+1,s+1);
  end
end

% Calculate the eta tilde matrix
Etat = zeros(N);
for n = N-1:-1:0
  for k = n:-1:0
    Acc = 0;
    for s = n+1:N-1
      Acc = Acc + Etat(n+1,s+1)*Etat(k+1,s+1)/Etat(s+1,s+1);
    end
    Etat(k+1,n+1) = Eta(k+1,n+1) - Acc;
  end
end

% Calculate the E matrix
E = eye(N);
for m = N-2:-1:0
  Tmp = eye(N);
  Tmp(m+1,m+2:N) = -Etat(m+1,m+2:N)./(diag(Etat(m+2:N,m+2:N)))';
  E = Tmp*E;
end
E = diag(1./diag(sqrt(Etat))) * E;
Et = inv(E);
```

Glossary

ADC	Analog to digital converter.
CGM	Channel gain measurement.
Cross talk	A disturbance caused by electric, magnetic, optic, acoustic or other means and originating from within the system itself.
DAC	digital to analog converter.
DWT	Discrete wavelet transform.
EUR	Euro.
Hadamard	A $N \times N$ matrix $\mathbf{H}$ is a Hadamard matrix if all entries are ± 1 and $\frac{1}{\sqrt{N}}\mathbf{H}$ is orthogonal.
IR	Acronym for 'infrared' and 'impulse response'.
JTF	Joint time-frequency.
LED	Light emitting diode.
LTT	Local trigonometric transform.
MSE	Mean square error.
Obtuse angle	An angle of between $\pi/2$ and π.
PCB	Printed circuit board.
Prolate	A spheroid which is given as revolution of an ellipse around its semimajor axis.
RST	Rudin-Shapiro transform.
Spheroid	The revolution of an ellipse around one of its semi-axes.
SS	Spread spectrum.
st	Steradian. The unit solid angle which cuts unit area from the surface of a sphere of unit radius centered at the vertex of the solid angle.
Unitary	A matrix is unitary if the adjoint equals the inverse. A real, unitary matrix is an orthogonal matrix.
WT	Wavelet transform.
WP	Wavelet packet.
WPT	Wavelet packet transform.

List of Figures

List of Tables

Bibliography

[1] N. Ahmed and K.R. Rao. *Orthogonal Transforms for Digital Signal Processing*. Springer-Verlag, New York, 1975.

[2] J.-P. Allouche and M.M. France. On an extremal property of the Rudin-Shapiro sequence. *Mathematika*, 32:33–38, 1985.

[3] P. Auscher, G. Weiss, and M.V. Wickerhauser. Local sine and cosine bases of Coifman and Meyer and the construction of smooth wavelets. In Charles K. Chui, editor, *Wavelets*, pages 237–256. Academic Press, Boston, MA, 1992.

[4] J. Beck. Flat polynomials on the unit circle. *Bull. London Math. Soc.*, 23:269–277, 1991.

[5] E. Beller. Polynomial extremal problems in L^p. *Proc. Amer. Math. Soc*, 30(2):249–259, 1971.

[6] G. Benke. Generalized Rudin-Shapiro systems. *J. Fourier Anal. Appl.*, 1(1):87–101, 1994.

[7] Berg and Reuss Christensen and Ressel. *Harmonic Analysis on Semigroups*, volume 100 of *Springer Graduate Texts in Mathematics*. Springer Verlag, 1984.

[8] J. Brillhart. On the Rudin-Shapiro polynomials. *Duke Math. J.*, 40:335–353, 1973.

[9] John Brillhart and Patrick Morton. Über Summen von Rudin-Shapiroschen Koeffizienten. *Illinois J. Math.*, 22(1):126–148, 1978.

[10] C.S. Burrus, R.A. Gopinath, and H. Guo. *Introduction to Wavelets and Wavelet Transforms. A Primer*. Prentice Hall, New Jersey, 1998.

[11] J. S. Byrnes. Quadrature Mirror Filters, Low Crest Factor Arrays, Functions Achieving Optimal Uncertainty Principle Bounds, and Complete Orthonormal Sequences – A Unified Approach. *App. and Comp. Harm. Anal.*, 1:261–266, 1994.

[12] J.S. Byrnes. On polynomials with coefficients of modulus one. *Bull. London Math. Soc.*, 9:171–176, 1977.

[13] J.S. Byrnes, I. Gertner, G. Ostheimer, and M.A. Ramalho. Discrete one dimensional signal processing apparatus and method using energy spreading coding, June 15, 1999. U.S. Patent no. 5,913,186.

[14] J.S. Byrnes, B. Saffari, and H.S. Shapiro. Energy Spreading and Data Compression Using the Prometheus Orthonormal Set. In J.M. Lervik and P. Waldemar, editors, *Digital Signal Processing Workshop Proceedings*, pages 0–12. IEEE, September 1996.

[15] S. Chen, C.F.N. Cowan, and P.M. Grant. Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. *IEEE Trans. Neural Net.*, 2(2):302–309, march 1991.

[16] T.S. Chihara. *An introduction to orthogonal polynomials*. Gordon and Breach Science Publishers, New York, 1978. Mathematics and its Applications, Vol. 13.
[17] C.K. Chui. *Wavelets: A Mathematical Tool for Signal Analysis*. SIAM, 1997.
[18] T.A.C.M. Claasen and W.F.G. Mecklenbrauer. The wigner distribution - a tool for time-frequency signal analysis. part i: Continuous-time signals. *Philips Journals of Research*, 35(3):217 – 250, 1980.
[19] T.A.C.M. Claasen and W.F.G. Mecklenbrauer. The wigner distribution - a tool for time-frequency signal analysis. part ii: Discrete-time signals. *Philips Journals of Research*, 35(4/5):276 – 300, 1980.
[20] T.A.C.M. Claasen and W.F.G. Mecklenbrauer. The wigner distribution - a tool for time-frequency signal analysis. part iii: Relations with other time-frequency signal transformations. *Philips Journals of Research*, 35(6):372 – 389, 1980.
[21] J.C. Clunie. The minimum modulus of a polynomial on the unit circle. *Quart. Jour. Math.*, 10(2):95–98, 1959. Oxford.
[22] A. Cohen, I. Daubechies, and P. Vial. Wavelets on the Interval and Fast Wavelet Transforms. *App. and Comp. Harm. Anal.*, 1:54 – 81, 1993.
[23] R. Coifman, F. Geshwind, and Y. Meyer. Noiselets. *App. and Comp. Harm. Anal.*, 10:27–44, 2001. Was available as preprint in 1994.
[24] J.W. Cooley and J.W. Tukey. An algorithm for the machine computation of complex fourier series. *Mathematics of Computation*, 19:297 – 301, April 1965.
[25] G.R. Cooper and C.D. McGillem. *Modern Communications and Spread Spectrum*. McGraw-Hill, New York, 1986.
[26] I. Daubechies. *Ten Lectures on Wavelets*, volume 60 of *CBSM-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, Pa., 1992.
[27] I. Daubechies. Orthonormal bases of compactly supported wavelets. II. Variation on a theme. *SIAM J. Math. Anal.*, 24(2):499–519, march 1993.
[28] U. Depczynski. Sturm-Liouville Wavelets. *Appl. Comp. Harm. Anal.*, 5(2):216–247, April 1998.
[29] U. Depczynski, K. Jetter, K. Molt, and A. Niemöller. The fast wavelet transform on compact intervals as a tool in chemometrics II. Boundary effects, denoising and compression. *Chemometrics and Intelligent Laboratory Systems*, 49(2):151–161, October 1999.
[30] R.C. Dixon. *Spread Spectrum Systems with Commercial Applications*. John Wiley & Sons, New York, 1994.
[31] P. Erdös. Some unsolved problems. *Michigan Math. J.*, 4:291–300, 1957.
[32] M. L. Fredman, B. Saffari, and B. Smith. Polynômes réciproques: conjecture d'Erdös en norme L^4, taille des autocorrélations et inexistence des codes de Barker. *C. R. Acad. Sci. Paris Sér. I Math.*, 308(15):461–464, 1989.
[33] M.J.E. Golay. Multislit spectrometry. *J. Optical Soc. Amer.*, 39:437–444, 1949.
[34] M.J.E. Golay. Complementary series. *IRE Trans.*, IT-7:82–87, 1961.
[35] K. Gröchenig. *Foundations of time-frequency analysis*. Birkhäuser Boston Inc., Boston, MA, 2001.

[36] P.C. Hansen. *Numerical aspects of deconvolution*. Preprint, december 1998.

[37] P.C. Hansen. Regularization Tools. A Matlab Package for Analysis and Solution of Discrete Ill-Posed Problems. Version 3.0 for Matlab 5.2., 1998. Technical Report IMM-REP-1998-6. Available at http://www.imm.dtu.dk/ ∼pch/Regutools/ regutools.html.

[38] G.H. Hardy and J.E. Littlewood. Some problems of Diophantine approximation: A remarkable trigonometrical series. *Proc. National Acad. Sc.*, 2:583–586, 1916.

[39] C. Herley, J. Kovačević, K. Ramchandran, and M. Vetterli. Tilings of the Time-Frequency Plane: Construction of Arbitrary Orthogonal Bases and Fast Tiling Algorithms. *IEEE Transactions on Signal Proccesing*, 41(12):3341 – 3359, december 1993.

[40] C. Herley and M. Vetterli. Orthogonal time-varying filter banks and wavelet packets. *IEEE Transactions on Signal Proccesing*, 42(10):2650 – 2663, october 1994.

[41] E. Hernández and G. Weiss. *A first course on wavelets*. CRC Press, Boca Raton, FL, 1996. With a foreword by Yves Meyer.

[42] N. Hess-Nielsen. *Time-Frequency Analysis of Signals Using Generalized Wavelet Packets*. PhD thesis, Aalborg University, Institute of Electronic Systems, June 1992.

[43] N. Hess-Nielsen. Control of frequency spreading of wavelet packets. *Applied and Computational Harmonic Analysis*, 1(2):157 – 168, March 1994.

[44] D. Hyder. Infrared Sensing and Data Transmission Fundamentals. http://www.web-ee.com/primers/files/an1016.rev0.pdf.

[45] A. Jensen and A. la Cour-Harbo. *Ripples in Mathematics - The Discrete Wavelet Transform*. Springer, Heidelberg Berlin, june 2001.

[46] J.-P. Kahane. Sur les polynomes a coefficients unimodulaires. *Bull. London Math. Soc.*, 12:321–342, 1980.

[47] S. Kay. A fast and accurate single frequency estimator. *IEEE Trans. on Acoust., Speech, and Signal Processing*, 12:1987–1990, Dec 1989.

[48] A. Kesteloot and C. L. Hutchinson. *The ARRL Spread Spectrum Sourcebook*. The American Radio Relay League, Newington, CT, 1991.

[49] J.D. Klein. Recursive single frequency estimation. *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001, Proceedings of.*, 5:3077 – 3080, 2001.

[50] T.W. Körner. On a polynomial of Byrnes. *Bull. London Math. Soc.*, 12:219–224, 1980.

[51] A. la Cour-Harbo and Jakob Stoustrup. Optimal threshold functions for active sensors. *Submitted to IEEE Trans. Indu. Elec., Special Issue on Intelligent Sensors*, 2003.

[52] P. G. Lemarié-Rieusset and G. Malgouyres. Support des fonctions de base dans une analyse multirésolution. *C. R. Acad. Sci. Paris*, t. 313, Série I, p. 377–380, 1991.

[53] J.E. Littlewood. On the mean values of certain trigonometric polynomials (ii). *Illinois J. of Math*, 6:1 – 39, 1962.

[54] J.E. Littlewood. On polynomials $\sum^n \pm z^m$, $\sum^n e^{\alpha_m i} z^m$, $z = e^{\theta i}$. *Journal London Math. Soc.*, 41:367–376, 1966.

[55] S. Mallat. Multiresolution approximation and wavelets. *Trans. Amer. Math. Soc.*, 315:69–88, 1989.

[56] H.S. Malvar. *Signal Processing with Lapped Transforms*. Artech House, Norwood, MA, 1992.

[57] M. Mendès France and G. Tenenbaum. Dimension des courbes planes, papiers pliés et suites de Rudin-Shapiro. *Bull. Soc. Math. France*, 109(2):207–215, 1981.

[58] Y. Meyer. *Ondelettes et opérateurs. I. Ondolette. II. Opérateur de Calderón-Zygmund. III. Opérateurs multilinéaires*. Hermann, Paris, 1990. English translation, Cambridge Univ. Press, London/New York, 1993.

[59] D.J. Newman. An l^1 extremal problem for polynomials. *Proc. Amer. Math. Soc.*, 16:1287 – 1290, 1965.

[60] J.-S. No, H. Chung, and M.-S. Yun. Binary pseudorandom sequences of period $2^m - 1$ with ideal autocorrelation generated by the polynomial $z^d + (z+1)^d$. *IEEE Trans. Inform. Theory*, 44(3):1278 – 1282, May 1998.

[61] J.-S. No, S.W. Golomb, G. Gong, H.-K. Lee, and P. Gaal. Binary pseudorandom sequences of period $2^n - 1$ with ideal autocorrelation. *IEEE Trans. Inform. Theory*, 44(2):814 –817, March 1998.

[62] J.-S. No and P.V. Kumar. A new family of binary pseudorandom sequences having optimal periodic correlation properties and larger linear span. *IEEE Trans. Inform. Theory*, 35(2):371 – 379, March 1989.

[63] F. Pedersen. *Joint time frequency analysis in digital signal processing*. PhD thesis, Aalborg University, The DSP Research Group, Dept. of Communication Technology, May 1997.

[64] S. Qian and D. Chen. *Joint Time-Frequency Analysis*. Prentice Hall, 1996.

[65] Alan Rogers. *Essentials of Optoelectronics With applications*, volume 4 of *Optical and Quantum Electronics*. Chapman & Hall, 1997.

[66] W. Rudin. Some theorems on Fourier coefficients. *Proc. Amer. Math. Soc.*, 10:855–859, 1959.

[67] Bahman Saffari. Une fonction extrémale liée à la suite de Rudin-Shapiro. *C. R. Acad. Sci. Paris Sér. I Math.*, 303(4):97–100, 1986.

[68] Bahman Saffari and Brent Smith. Sur une note récente relative aux polynômes à coefficients ± 1 et à la conjecture d'Erdös. *C. R. Acad. Sci. Paris Sér. I Math.*, 310(7):541–544, 1990.

[69] I.W. Selesnick. The slantlet transform. *IEEE Trans. Signal Proc.*, 47(5):1304 – 1313, May 1999.

[70] H.S. Shapiro. Extremal problems for polynomials and power series. Master's thesis, Massachusetts Institute of Technology, may 1951.

[71] Simon, Omura, Scholtz, and Levitt. *Spread Spectrum Communications Handbook*. McGraw-Hill, New York, 2 edition, 1994.

[72] J. Spencer. Six standard deviations suffice. *Trans. Amer. Math. Soc.*, 289(2):679–706, June 1985.
[73] STC. Sensor foresight report. Technical report, Sensor Technology Center A/S, November 2001. www.sensortec.dk.
[74] G. Szego. *Orthogonal Polynomials*. Amer. Math. Soc., Providence, RI, 4th edition, 1975.
[75] M. Taghavi. An estimate on the correlation coefficients of the Rudin-Shapiro polynomials. *Iranian J. Sci. Tech.*, 20(2, Trans. A Sci.):235–240, 1996.
[76] M. Taghavi. Upper bounds for the autocorrelation coefficients of the Rudin-Shapiro polynomials. *Korean J. Comput. Appl. Math.*, 4(1):39–46, 1997.
[77] A.S. Tanenbaum. *Computer Networks*. Prentice Hall, third edition, 1996.
[78] F. Trèves. *Topological vector spaces, distributions and kernels*. Academic Press, New York, 1967.
[79] M. Vetterli and J. Kovačević. Time-varying modulated lapped transforms. In *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 481–485. IEEE, 1993.
[80] M. Vetterli and J. Kovačević. *Wavelets and subband coding*. Prentice-Hall, 1995.
[81] A.J. Viterbi. *CDMA: Principles of Spread Spectrum Communications*. Addison-Wesley, Reading, MA, 1995.
[82] A.J. Viterbi. Spread spectrum communications: myths and realities. *IEEE Communications Magazine*, 40(5):34 – 41, May 2002. 50th Anniversary Commemorative Issue (first published in IEEE Communications Magazine, May 1979).
[83] M.V. Wickerhauser. *Adapted Wavelet Analysis from Theory to Software*. A K Peters, May 1994.
[84] T.P. Zielinski. Joint time-frequency resolution of signal analysis using gabor transform. *IEEE Trans. Instrumentation and Measurement*, 50(5):1436 – 1444, October 2001.
[85] A. Zygmund. *Trigonometrical series*. Warsaw, 1935.

Index

History

July 1st 2002

Original version.

July 8th 2002

- Introduction finished.
- Discussion and Future Work chapter included.
- Section 4.9 rewritten slightly. Still not completely finished.
- More references on SS systems included (suggested by Mr. Thuillard).
- A series of minor corrections.

August 8th 2002

- Section 4.9 and 5.3 significantly changed. The plots in 4.9 have been updated and derivations for stochastic signals y_0 is included. The derivations for the second validation method is also included. The test results in 5.3 are also changed, though only slightly, to fit the rewriting.
- Index included.
- History of spread spectrum sequences finished.
- Font in headings changed from Computer Modern to Helvetica.
- Layout has been adjusted for final version printing.